Distributed Localisation Algorithm for Wireless Ad Hoc Networks of Moving Nodes

Milan Cvjetković

Supervisor: Dr Veselin Rakočević

Department of Electrical and Electronic Engineering

City, University of London

This dissertation is submitted for the degree of

Doctor of Philosophy

July 2019

Declaration

I hereby grant powers of discretion to the University Librarian to allow the thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement. This declaration shall follow the acknowledgements.

Milan Cvjetković

July 2019

Acknowledgements

I would like to thank Dr Veselin Rakočević, my supervisor, for continuous encouragement and guidance throughout my PhD research. Also, I would like to express gratitude to my parents and sister for their unconditional support and endless patience.

Abstract

Existing ad hoc network localisation solutions rely either on external location references or network-wide exchange of information and centralised processing and computation of location estimates. Without these, nodes are not able to estimate the relative locations of other nodes within their communication range. This thesis defines a new distributed localisation algorithm for ad hoc networks of moving nodes. The *Relative Neighbour Localisation* (RNL) algorithm works without any external localisation signal or systems and does not assume centralised information processing. The idea behind the location estimates produced by the RNL algorithm is the relationship between the relative locations of two nodes, their mobility parameters and the signal strengths measured between them. The proposed algorithm makes use of the data available to each node to produce a location estimate. The signal strength each node is capable of measuring is used as one algorithm input. The other input is the velocity vector of the neighbouring node, composed of its speed and direction of movement, which each node is assumed to periodically broadcast. The relationship between the signal strength and the mobility parameters on one, and the relative location on the other side can be analytically formulated in an ideal case. The limitations of a realistic scenario complicate this relationship, making it very difficult to formulate analytically. An empirical approach is thus used. The angle and the distance estimates are individually computed, together forming a two-dimensional location estimate. The performance of the algorithm was analysed in detail using simulation, showing a median estimate error of under 10 m, and its application was tested through design and evaluation of a distributed sensing coverage algorithm, showing RNL location estimates can provide 90% of the coverage achievable with true locations being known.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

A-GPS	Assisted Global Positioning System
AoA	Angle of Arrival
AP	Access Point
BS	Base Station
CID	Cell ID
CRS	Cell-specific Reference Signal
D2D	Device-to-Device
DSSS	Direct Sequence Spread Spectrum
E-CID	Enhanced Cell ID
FCC	Federal Communications Commission
GLONASS	GLObal NAvigation Satellite System
GNSS	Global Navigation Satellite System
GPS	Global Positioning System

GSM	Globas System for Mobile communication
HSUPA	High Speed UPlink Packet Access
IoT	Internet of Things
LMU	Location Measurement Unit
LOS	Line Of Sight
LTE	Long Term Evolution
MAC	Medium Access Control
MS	Mobile Station
NLOS	Non-Line Of Sight
OFDM	Orthogonal Frequency Division Multiplexing
OTDoA	Observed Time Difference of Arrival
PRS	Positioning Reference Signal
RFR	Random Forest Regression
RNL	Relative Neighbour Localisation
RSS	Received Signal Strength
RTT	Round Trip Time
RXLEV	Received signal strength (in GSM)
SINR	Singnal to Interference plus Noise Ratio
TA	Timing Advance

TDMA	Time-Division Multiple Access
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UTDoA	Uplink Time Difference of Arrival
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Overview

The development of wireless communication technologies in recent decades has led to the present day ubiquity of wireless networks. The technological advances have not only made Internet access available from almost any place at any time, they are also constantly making the design of smaller, more powerful and more autonomous devices possible. People are no longer the only users of telecommunication networks, this role is rapidly being taken by the devices themselves. The emergence of concepts like *device-to-device* (D2D) communication and the *Internet of Things* (IoT) has led to an increasing number of devices joining local or global wireless communication networks, enabling them to perform their functions more efficiently.

Local networks of devices operating on a common task or exchanging information are used for many different applications. “Smart” appliances in “smart” homes, autonomous vehicles or different types of sensors are being connected among themselves. *Infrastructureless*, or *ad hoc*, networks are being used for the purpose of connecting these devices, as in order to operate efficiently the devices do not necessarily need Internet connectivity, and the focus is on the local information exchange. Unlike *infrastructure-based* networks which are sup-

ported by a number of *base stations* (BS) or *access points* (AP) which provide connectivity to all users, ad hoc networks are formed and maintained by the nodes themselves.

In addition to the inherent challenges all wireless communication systems face, the infrastructureless nature of ad hoc networks exposes these networks to further issues. The connectivity of nodes relies on the topology of the nodes surrounding them, and node mobility can cause frequent and significant changes in this topology. Without a fixed network infrastructure, external localisation systems like *Global Positioning System* (GPS), or specific ranging hardware in place, estimating locations of moving nodes in ad hoc networks becomes a challenge.

This research is motivated by the fact that ad hoc networks of moving nodes are often used for location-related group tasks. If the nodes in such networks are unable to obtain location information from external sources, their efficiency at performing task at hand is challenged. Providing nodes with general information about the physical topology of their neighbours can help increase this efficiency, regardless of the inaccuracy of exact location estimates produced.

Existing solutions in the field of ad hoc network localisation either assume that some sort of external localisation is present, in form of *beacon*, or *anchor*, nodes which are aware of their location and are used as references, or that the information from measurements between nodes across the whole network is exchanged and centrally processed. This thesis, on the other hand, presents a localisation algorithm which is distributed, i.e. each moving node in an ad hoc network can run the algorithm itself to produce estimates of its neighbours' locations. No presence of external localisation is assumed, and the exchange of information between nodes is limited to sharing their mobility parameters.

The goal of the proposed *Relative Neighbour Localisation* (RNL) algorithm is to provide a location estimate which can be used in location-based cooperative networks, based on the information available to each node. The basis of the proposed approach is an intuitive

relation between two nodes' mobility parameter, relative position and the signal strength between them. The information used as algorithm input is *unreliable* as a consequence of signal propagation effects, and the algorithm aims to provide an insight into the relative location of the neighbouring node, useful to a location-related application, rather than exact location of it, as high level of accuracy based on the data available is difficult to achieve.

The designed localisation algorithm can be used in any ad hoc network of moving nodes. As it does not aim to provide the nodes with exact location of their neighbour, its contribution is in the fact that it is able to provide a general idea of the physical topology of the neighbouring nodes, which can be used in situations where other localisation sources are unavailable and the nodes forming the network are working on a location-related task. In such scenarios, having information about the general location, or direction, in which the neighbours are located can be beneficial for efficient task completion (in comparison to a case in which no location information is present).

The performance of the RNL location estimates has been evaluated using simulation. In order to ensure the simulation was realistic, a set of tests was performed in an experimental testbed which was developed. The models used in the simulation were designed and calibrated using the measurements obtained in the experimental tests. The RNL algorithm was shown to be able to produce median location estimate error of under 10 m. An application scenario was also designed, in which the RNL location estimates were used in a distributed sensing algorithm, showing that the coverage obtained based on the RNL location estimates was over 90% of the one achievable if the information about the true locations of neighbouring nodes were available.

1.2 Research Contributions

In light of the existing gap in the field of localisation for ad hoc networks of moving nodes, without the support of external signals or systems, the main contributions of this work may be defined as:

- Formulation of a principle for distributed localisation in ad hoc networks of moving nodes. This principle is based on the relationship between the trends in measurements of the received signal strength from a neighbouring node over time and the neighbouring node's mobility parameters (speed and direction of movement) on one side, and the relative location of that neighbouring node on the other.
- Design of a distributed localisation algorithm for ad hoc networks of moving nodes without the use of any external localisation signals or systems. The proposed algorithm is designed in a way that allows nodes to individually produce estimates of neighbouring nodes location, without any centralised knowledge.
- Application of the localisation algorithm in a distributed sensing coverage scenario.

1.3 Publications

The following papers have been published throughout this PhD study:

- Cvjetkovic, M., & Rakocevic, V. (2016). Alternative topology construction for cooperative data distribution in mobile ad hoc networks. Paper presented at the 2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 171-176.
- Cvjetkovic, M., & Rakocevic, V. (2017). Relative localisation algorithm for neighbour classification in ad hoc networks of moving robots. Paper presented at the Proceedings

of the First ACM International Workshop on the Engineering of Reliable, Robust, and Secure Embedded Wireless Sensing Systems, 46-53.

1.4 Thesis Structure

The rest of this thesis is structured as follows: the second chapter provides an overview of localisation in wireless networks. The need for and the applications of localisation are presented, followed by the description of the basic principles of different types of localisation algorithms. Existing solutions for localisation in infrastructure (cellular and WLAN) and infrastructureless networks are reviewed in the second chapter as well.

The third chapter defines the proposed RNL algorithm for distributed localisation in ad hoc networks of moving nodes. Theoretical principles of localisation in an ideal case are presented, followed by a discussion on the limitations existing in a realistic scenario, and their implications on the application of the proposed localisation approach. Finally, the design of individual elements and procedures of the RNL algorithm is presented in detail.

Chapter four presents the evaluation of the performance of the RNL algorithm. It starts with a description of the experimental setup used to gather measurements for designing realistic models used in simulations. The accuracy results of simulation tests of different types of location estimates are then presented. Finally, a potential application of the estimates produced by the RNL is tested in a scenario where distributed sensing coverage is used, examining the effects of the inaccuracy of RNL location estimates.

The final chapter concludes the thesis, providing a summary of the proposed solution and its performance, as well as a discussion about possible directions of future work.

Chapter 2

Localisation in Wireless Networks

This chapter presents an overview of different types of wireless networks and the need for localisation in them. The basics of different localisation approaches are then introduced, after which an overview of the existing localisation solutions for infrastructure-based and infrastructureless wireless networks is presented. It concludes with the definition of the problem this thesis proposes a solution for.

2.1 Wireless Networks and the Need for Localisation

2.1.1 Types of wireless networks

Development of wireless communication using radio waves has made computers and people (commonly called *nodes* or *users*) able to communicate without having a physical connection to the underlying network. Different systems relying on wireless communication have been developed, with new applications and services being provided to users constantly, and users being able to access them from wider areas and almost at all times, even when on the move. The shift in communication and Internet access paradigms has, on the other hand, introduced a number of new challenges, the majority of which the traditional, wired, communication

systems have not had to deal with, with their prominence varying depending on the type of wireless system in use.

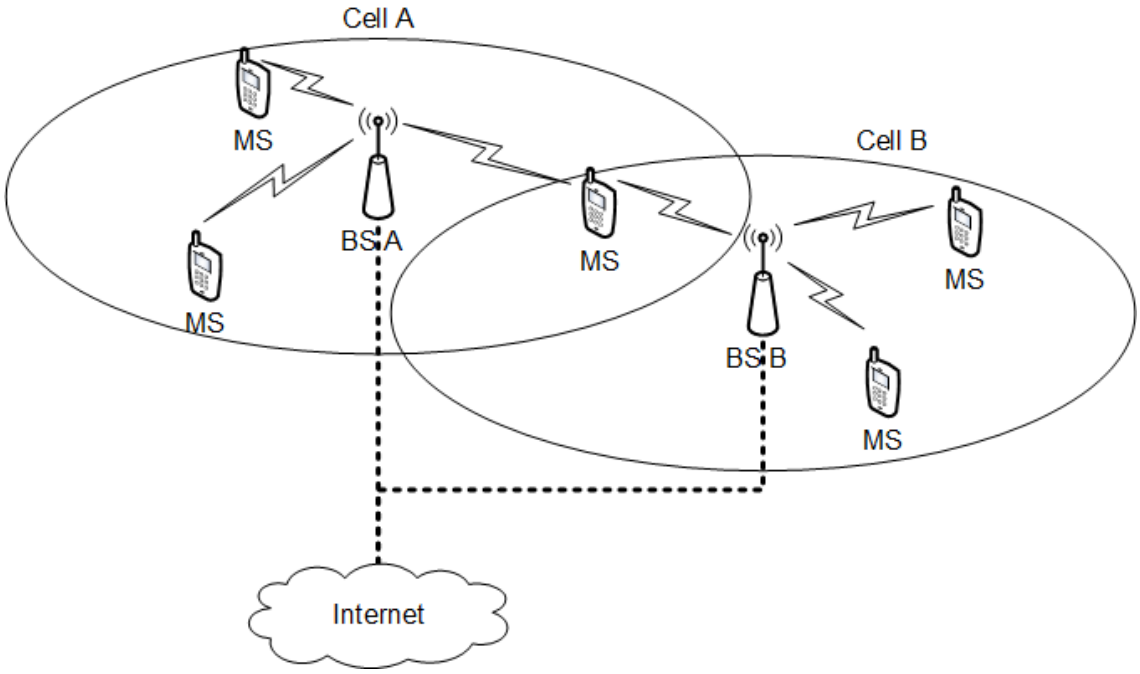


Fig. 2.1 Wireless cellular network

Wireless communication systems can exist as parts of a bigger network with a fixed core, in which case they are used for extending the range and widening the number of users able to use its services. This is the case in *infrastructure-based networks* which are based around an access point (AP), or a base station (BS). The AP serves as the gateway for wireless users, providing access to the rest of the network. It usually has a direct wired connection to the core, but sometimes, in remote areas or for other practical reasons, a *point-to-point* link is used to connect it to the rest of the network. Figure 2.1 shows an example of a cellular network infrastructure. The main sources of issues infrastructure-based wireless networks face are the user mobility and the effects signal propagation has on signal stability. Specific routing protocols and handover procedures need to be in place for uninterrupted connectivity in situations when mobile users switch from the coverage area of one access point to the coverage area of another one. Power control mechanisms are needed for the user equipment

(UE) (or mobile station (MS)) to account for the user moving closer or further away from the AP, and adaptive modulation schemes are used for enabling connectivity in conditions of varying signal strength. It can be concluded that even with the existence of infrastructure in a wireless network, there is a number of issues arising from user mobility as well as wireless signal propagation effects.

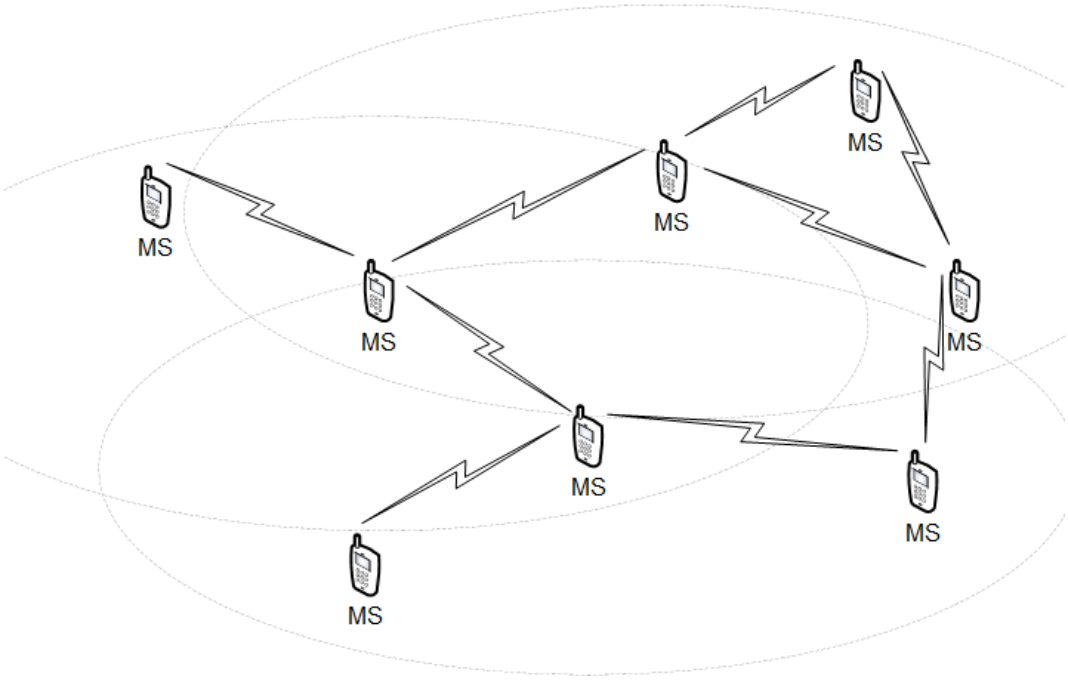


Fig. 2.2 Infrastructureless radio network

The other type of wireless communication networks are *infrastructureless*, or *ad hoc* networks. These networks are characterised by self-organisation, as there is no infrastructure support. The nodes form the network by cooperating among themselves, and all communication is performed directly between the sending and the receiving nodes. Figure 2.2 represent an example of on infrastructureless network. The absence of any centralised infrastructure means that these networks provide greater flexibility and easier deployment, but it also exposes such networks to additional challenges, on top of the ones inherent to all wireless communication systems. In the case of mobile nodes, the nodes can move in and out of communication range of other nodes, sometimes becoming completely unreachable.

Different routing (*mesh*), *cooperative networking* and *opportunistic networking* protocols are used for ad hoc networks with interrupted or limited connectivity.

Wireless *sensor* networks are networks of nodes with sensing capability. The nodes all measure a certain physical quantity at their location, and the network is then used to distribute these measurements to designated collection points. Wireless sensor networks are usually deployed using an ad hoc architecture, as it provides greater flexibility and is simpler to set up. The nodes in such networks can be either static or mobile, depending on the application they are designed for.

2.1.2 Localisation fundamentals

As previously stated, wireless technology has enabled network nodes to establish communication without a physical connection to other network elements, and more importantly, to remain connected while on the move. Node location thus became a variable and became an important parameter in many different applications and services. This section represents an overview of the basic terms and principles used in the field of *determining node location* in wireless networks – the process of *localisation*. It is important to note that all localisation techniques operate with a certain error, so the term *location estimate* will be used to refer to the node location as determined by the localisation technique.

Node location may be needed for different *purposes*, which may define the parameters of the localisation technique such as precision, delay, or whether the node or the network are performing the localisation. Based on the goal towards which a node's location is estimated, there are three main categories:

- Node location can be the *purpose in itself*, i.e. positioning or navigation applications; emergency services.
- Node location is used for a *network application or protocol*, i.e. location-based routing, coverage or sensing.

- Node location is used for a *higher layer application*, i.e. location-based services such as locating nearest businesses of the desired type, asset management, advertising or gaming.

Positioning and navigation applications have become ubiquitous in the present day, from end users trying to determine their location or finding their way to a certain location, to use in autonomous vehicles, logistics and other industries. From the network's perspective, in these scenarios, the goal is just providing a location estimate. What this information is later used for is of no interest to the network. Another very important use of location and a significant driving force of research in this field is in emergency scenarios. The importance of locating the users in need of help has led to modern public communication systems having the capability of estimating user location incorporated in their standards.

Network applications and protocols relying on node location are designed to maintain connectivity or enable location-related data distribution efficiency in networks of mobile users with dynamic topology. In cellular radio networks, node location may be used for making handover decisions for high-speed moving nodes. Ad hoc and relay networks may use node location for routing purposes, speeding up the process of route discovery or proactively detecting changes in topology. Wireless Sensor Networks rely on node location for coverage control – maintaining the whole area of interest covered as well as gathering data from the sensors in the most efficient way. In all of the scenarios listed above, it is the network itself which requires the node locations, in order to be able to provide full functionality.

With an increase in usage of wireless communication by mobile users the need for providing these users with content and services related to their location has arisen. User localisation made it possible for users in specific locations to be "targeted" by certain content, to have access to different content based on their location, or to receive recommendations on different nearby services. Location-based services have also found usage in healthcare,

transport, logistics and other fields. In such scenarios, the network helps determine nodes' locations and then passes the information on to the service or application requesting it.

Node locations may be estimated using different *reference systems*, depending on the localisation technique used. Here, two possible approaches exist:

- *Absolute* localisation – a common coordinate system exists which all location estimates are expressed in.
- *Relative* localisation – location estimates are made in relation to another device whose location is known.

The most prominent examples of absolute localisation are satellite positioning systems like the *Global Positioning System* (GPS) [1], *Global Navigation Satellite System* (GLONASS) [2] and *Galileo*, collectively referred to as *Global Navigation Satellite Systems* (GNSS) [3]. These systems consist of constellations of satellites with well-known locations in a universal coordinate system. The satellites are equipped with highly stable synchronised atomic clocks and are constantly transmitting their time and location information. If the user receives these signals from enough different satellites it is then capable to solve a set of equations resulting in its position in the universal coordinate system, along with its internal clock time deviation, which means that these types of systems can also be used for time synchronisation of the user equipment.

In contrast, relative localisation techniques use either a set of devices with known fixed locations or mobile devices capable of determining their absolute location, making their position always known to others. These nodes are called *anchors*, *beacons* or *landmarks*, and are used to define a "local" coordinate system. Using one of the different techniques described below, other nodes can determine their location within the established reference system, and, if necessary and possible, translate this relative location into an absolute one.

2.2 Localisation Parameters

Localisation techniques may use different types of *parameters*, which can be split into two categories:

- *Range-based* – physical characteristics of the received signal are analysed.
- *Range-free* – topology formed by links between nodes is analysed.

2.2.1 Range-based localisation

Localisation methods based on signal properties analysis, also called *range-based* methods, consider physical properties of the received signals and try to directly translate measurements of these properties into the distance or angle estimates. The errors in these estimates depend on the level of precision of the measurements. The signal properties used are *signal strength* – in form of *received signal strength* (RSS) parameter, *time of arrival* (ToA) or *time difference of arrival* (TDoA), or *angle of arrival* (AoA). In order to perform precise measurements of these parameters, specific equipment is usually needed, which increases the cost and complexity of such methods. On the other hand, using such precise measurements increases the localisation accuracy. Range-based methods approach the localisation problem from a geometrical perspective, and depending on which of the signal parameters is used in the problem definition they can be defined as:

- *Trilateration* – based on RSS or ToA
- *Multilateration* – based on TDoA
- *Triangulation* – based on AoA

2.2.2 Range-free localisation

Localisation methods based on analysing the connectivity and topology of the network, rather than particular signal properties, are called range-free localisation methods. These methods are either relying on information from individual nodes' neighbourhood lists or routing tables, or they are based on matching patterns of RSS measurements from multiple sources. It is worth noting that the RSS parameter is not used to estimate distances in pattern matching algorithms, which is why these methods can still be considered range-free. No precise measurement of physical signal properties are needed for range-free localisation, so no additional equipment is needed, as the algorithms can be implemented in any node capable of wireless communication. This makes the range-free approach cheaper and less complex to implement, but the accuracy they produce is usually lower than the range-based ones.

2.3 Localisation Methods

2.3.1 Trilateration

This range-based method determines the unknown user location using distance estimates. In three-dimensional space, the estimated distance from a transmitter with known location defines a sphere with the centre in that point and radius equal to the distance estimate, on the surface of which the user is located. By acquiring multiple (three or more) distance estimates to transmitters with known locations, the unknown location is calculated as the intersection of these spheres. The transmitting and receiving roles are interchangeable in this procedure, so the location is not necessarily calculated by the node with an unknown location, but may be calculated by the network itself, gathering the measurements from different receivers positioned in known locations.

In a simplified model, the process of trilateration assumes solving the following system of equations:

$$\begin{aligned} r_1 &= \sqrt{(x_1 - x)^2 + (y_1 - y)^2 + (z_1 - z)^2} \\ r_2 &= \sqrt{(x_2 - x)^2 + (y_2 - y)^2 + (z_2 - z)^2} \\ &\dots \\ r_n &= \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2}, \end{aligned} \tag{2.1}$$

where r_i is the distance between the transmitter and the receiver , x_i, y_i, z_i are known coordinates of transmitter i, n is the number of available distance estimates, and x, y, z are the unknown coordinates of the user. Figure 2.3 shows an example of trilateration, using three reference points and the estimated distances to them.

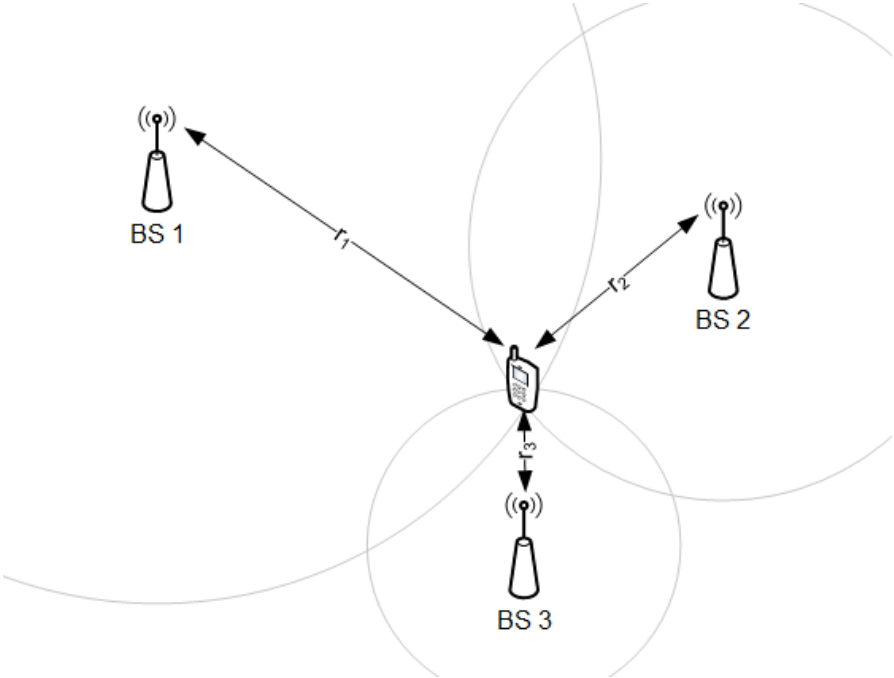


Fig. 2.3 Trilateration

Distances can be estimated from the received signal strength (RSS) if the transmitting power is known, which is a less precise option as signal propagation effects cause fading,

resulting in a random variation of signal strength at the receiver. Another way to estimate distances is from the time of arrival (ToA) of the signal, which is a more precise method, but increases complexity unless the transmitter and receiver clocks are synchronised.

When RSS is used, or ToA measurements are used, with system-wide time synchronisation in place, the minimal number of distance estimates from known locations needed is equal to the dimensionality of the space in which the location is being determined, d, so $n \geq d$ needs to hold for successful trilateration.

In systems where there is no time synchronisation and timing parameters are used, as is the case with GNSS, the number of unknowns in the system is one greater than the number of dimensions [4]. The additional unknown is the clock offset, τ, of the receiver, which is computed at the same time as the spatial coordinates. The offset of transmitter i, τ_i, is considered to be known. For such systems, $n \geq d+1$ needs to hold for trilateration to work. If that is the case, the concept of *pseudorange* needs to be introduced [4] and is defined as the product of the *difference in time the signal is transmitted, according to the transmitter clock, and the time the signal is received, according to the receiver clock* and the speed of light, c. Observed pseudorange from transmitter i can be expressed as:

$$\begin{aligned} P_i(t) &= ((t+\tau) - (t_i+\tau_i))\, c \\ &= (t-t_i)\, c + c\tau - c\tau_i \\ &= \rho^i(t,t_i) + c\tau - c\tau_i, \end{aligned} \tag{2.2}$$

where $\rho_i\left(t,t^i\right)$ is the distance between the receiver at time t, the time signal is received, and transmitter i at time t_i, the time signal is transmitted, and is calculated as:

$$\rho_i(t,t_i) = \sqrt{(x_i(t_i) - x(t))^2 + (y_i(t_i) - y(t))^2 + (z_i(t_s) - z(t))^2}. \tag{2.3}$$

A simplified system of pseudorange equations can now be derived as:

$$\begin{aligned} P_1 &= \sqrt{(x_1-x)^2+(y_1-y)^2+(z_1-z)^2}+c\tau-c\tau_1 \\ P_2 &= \sqrt{(x_2-x)^2+(y_2-y)^2+(z_2-z)^2}+c\tau-c\tau_2 \\ &\dots \\ P_n &= \sqrt{(x_n-x)^2+(y_n-y)^2+(z_n-z)^2}+c\tau-c\tau_n. \end{aligned} \tag{2.4}$$

In many cases, especially in GNSS, more than the minimum needed number of distance estimates is available. If that is the case, it is possible to use the over-determination of the system to increase the precision of the location estimate by means of least squares method as described in [5].

2.3.2 Multilateration

This range-based method is based on the differences between distance estimates. In three-dimensional space, the difference in estimated distances between two transmitters with known locations and the user with an unknown location defines a hyperboloid whose focal points are located in the two transmitters, on the surface of which the user is located. By acquiring multiple differences in distances between pairs of transmitters with known locations (three or more) the unknown location is derived as the intersection of the hyperboloids defined by them. Again, the devices which location is known may also be receivers, and the user with an unknown location may be a transmitter of the signal used to estimate the difference in distance, allowing for this procedure to be performed by either the user itself or the network.

A system of equations of the following form is obtained using the multilateration procedure [6]:

$$r_{i,j} = \sqrt{(x_i-x)^2+(y_i-y)^2+(z_i-z)^2}-\sqrt{\left(x_j-x\right)^2+\left(y_j-y\right)^2+\left(z_j-z\right)^2}, \tag{2.5}$$

where x_i, y_i, z_i are known coordinates of transmitter i, x_j, y_j, z_j are known coordinates of transmitter j, x, y, z are unknown coordinates of the user, and $r_{i,j}$ is the difference between the distances between the user and transmitters i and j. Figure 2.4 shows an example of multilateration, using distance differences between two pairs of reference points.

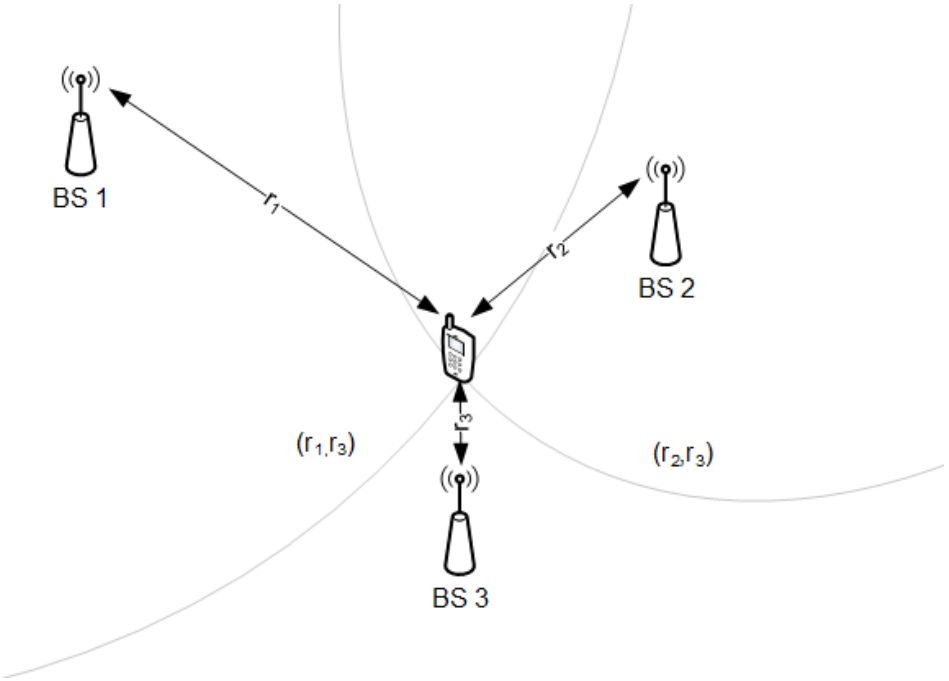


Fig. 2.4 Multilateration

The difference in distances between the user and two transmitters with known locations is calculated using the TDoA parameter. When calculating the difference in the times of signal arrivals from two different transmitters there is no need for the user clock to be synchronised to the transmitter clocks, as the user does not need to know the exact time the signals were sent. This makes the TDoA method much less complex than the ToA method used in trilateration.

If the transmitters are synchronised among themselves, the minimal number of pairs of transmitters for which differences in distances are estimated through the TDoA parameter needs to be greater or equal to the number of dimensions of the space in which the user location is estimated, resulting in three transmitters for both two-dimensional and three-

dimensional localisation. In practice, if there is no absolute synchronisation between the transmitters, one additional transmitter needs to be used as a time reference point, making four the minimal number of transmitters necessary for three-dimensional localisation.

Similar to trilateration, the availability of more than the minimal number of TDoA estimates needed can help increase the accuracy of the location estimate, discard an ambiguous solution, or help detect an error in one of the estimates. An extra reference point may intentionally be used with the purpose of simplifying the process of solving the system of multilateration equations, given the complexity induced by the square root terms [6].

2.3.3 Triangulation

This range-based localisation method utilises measurements of the angles at which the signal from the transmitter with unknown location (the user) arrives at receivers with known locations [7]. A known direction of arrival of the signal defines a half-line with the initial point in the receiver, and the transmitter is located somewhere along the half-line. Obtaining two or more angle of arrival (AoA) measurements defines the unknown location of the transmitter in the intersection of the half-lines. In two dimensions, each AoA measurement produces the following equation:

$$\theta_i = arctan\left(\frac{y - y_i}{x - x_i}\right), \tag{2.6}$$

where x_i and y_i are the known coordinates of the receiver i, θ_i is the measured angle of arrival of the signal coming from the transmitter with unknown location coordinates x and y. Figure 2.5 shows an example of triangulation, using AoA measurements of signals received from three different transmitters.

Precise measurements of the angle from which a signal arrives are possible using antenna arrays, making this method relatively expensive and impractical to implement on the user side.

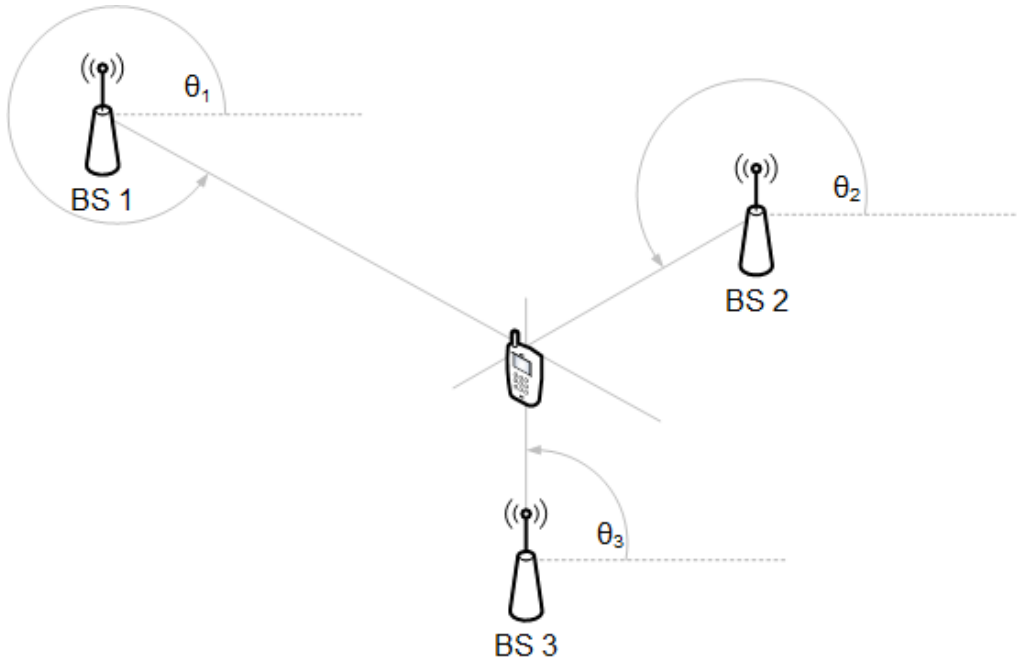


Fig. 2.5 Triangulation

This is why most of the applications of triangulation have the angle of arrival measurement performed at the fixed receivers at known locations, although some implementations may work with reversed roles, with the reference points acting as transmitters, and the AoA measurements performed by the user equipment. More than two reference points are often used, to reduce the effects of errors made in individual AoA measurements. Additional complexity is introduced with increasing the number of spatial dimensions in which angulation is performed, as determining an unknown location in three-dimensional space assumes detecting both the azimuthal angle θ and the polar angle ϕ, so an additional equation is produced by each AoA measurement:

$$\phi_i = arccos\left(\frac{z-z_i}{r}\right), \tag{2.7}$$

where:

$$r = \sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}. \tag{2.8}$$

2.3.4 Connectivity based methods

These methods rely on the *hop-count* metric – the number of nodes through which a packet travels from node *A* to node *B*. This metric can be assumed to indicate the distance between two nodes, although it may be largely influenced by node density, and may be used in two different ways: to determine the unknown location of the node based on its proximity to anchor nodes whose location is known; or to combine such knowledge from all nodes in the network and compute the overall physical topology of the network. Different algorithms, which are reviewed in the next section, are used to make the final estimate in the anchor proximity approach, as well as for combining the hop-count information from multiple nodes in order to minimise errors.

2.3.5 Pattern matching methods

These methods use the RSS measurements from multiple beacons to determine the unknown node location. The signal strength measurements are not used as indicators of actual distance but as characteristics of different parts of the area in which the network is deployed. Pattern matching can be done based on either probabilistic propagation models or empirical measurement databases which is why the solutions taking the latter approach are often called *fingerprinting* methods. A fingerprint is represented by a vector containing RSS measurements for a certain point in the area of interest. This is why such methods need an offline phase first, a phase in which data is gathered, measurements are taken with a predetermined resolution, and are stored in a database. The idea behind fingerprinting localisation methods is for the node with an unknown location to form a vector of measured signal strength values from all beacons in range, which can then be compared with the existing database. The fingerprint (or multiple fingerprints) which is the best match to the fingerprint reported by the user is found, and the location associated with it is returned as the location estimate. Possible

approaches to pattern matching in different types of networks are described in greater detail in the next section.

2.4 Existing Localisation Solutions in Radio Networks

2.4.1 Infrastructure-based networks

Localisation of users in infrastructure-based networks is done using both the range-based and range-free methods. The choice of localisation techniques depends on the type of network and the limitations imposed by the technology used in different networks. Cellular networks are equipped with the type of infrastructure which supports precise measurements and synchronisation may be present, so lateration techniques are often used. On the other hand, Wireless Local Area Networks (WLANs) are based on infrastructure which does not usually allow for such precise measurement, so pattern matching is the most common type of localisation algorithm used in these networks. As the networks evolved, so did the localisation techniques with them, adapting to the changes in communication technology, which resulted in many different implementations of the principles described in the previous section.

The main driving force behind localisation in cellular networks is the need for accurate location estimates in emergency situations, especially when users are located indoor, as GNSS systems tend to provide lower accuracy in such environments. Accuracy and availability requirements are usually defined by government bodies, such as the American *Federal Communications Commission* (FCC). The latest set of requirements issued by the FCC can be found in [8], where it is stated that network operators should provide 50 metres x, y accuracy for a steadily increasing percentage (40-80%) of emergency service calls over the period of 2 to 6 years. In wireless networks, because of their relatively low price and public availability,

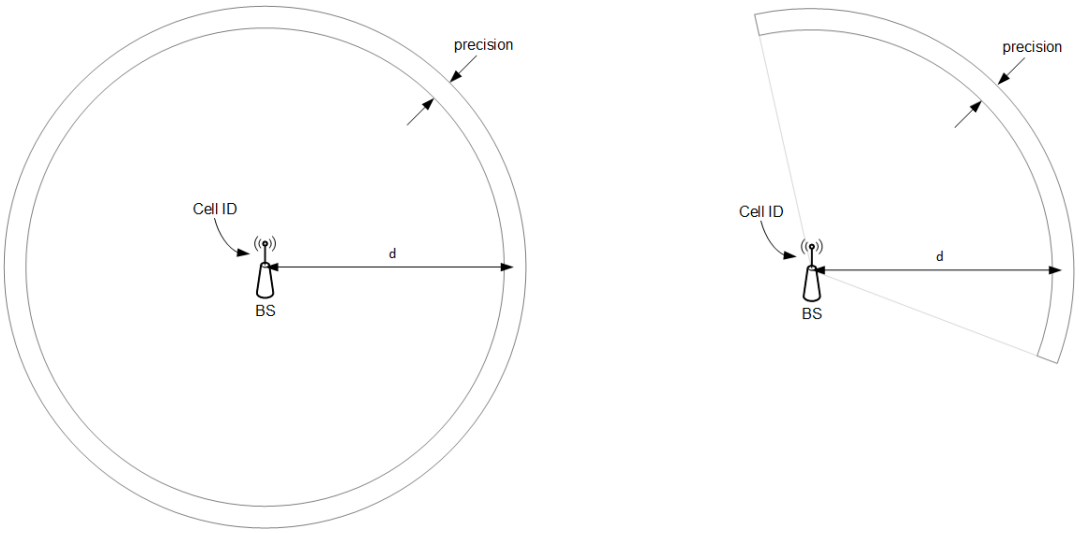
localisation has found its main use in remote healthcare, security and asset management [9], as well as in the gaming and augmented reality-based industries.

Range-based localisation in cellular networks

The cover term for localisation methods using the known locations of the BSs as the base for determining user location is *Cell-ID* (CID) (defined in [10]) in GSM (*Global System for Mobile communications*) networks, and *Enhanced Cell-ID* (E-CID) in UMTS (*Universal Mobile Telecommunications System*) and LTE (*Long Term Evolution*) networks (defined respectively in [11], [12]). Ranging methods are widely used in public mobile networks, regardless of the network generation, and most lateration techniques are implemented under the CID and E-CID names.

Cell-ID represents the identification number of the cell currently serving the MS. It may be used as a parameter for localisation purposes on its own [13], in combination with range estimates of lower [14, 15] or higher accuracy [16, 17]. On top of these basic GSM implementations, additional algorithms and procedures have been applied in UMTS networks to improve accuracy and availability by managing channel utilisation [18] or forcing handover procedure [19–21]. Further improvements came with LTE networks, as localisation-specific signals [22, 23], multiple propagation detection [24], and device collaboration [25] have been introduced. The locus of the location estimate produced using the information from a single BS CID and a range estimate is shown in Figure 2.6, with both the omnidirectional and sector antennas cases. The precision of the range estimate defines the thickness of the circular ring or ring sector. Regardless of the fact all ranging solutions share the same mathematical principles, the technology used to obtain location estimates is the most significant factor in defining their accuracy.

The most basic localisation method using the knowledge of the serving cell is just assuming the MS is located at the serving BS coordinates. An experimental study of



(a) Omni-directional antenna (b) Sector antenna

Fig. 2.6 Localisation using cell-ID and ranging

this approach is presented in [13]. Although extremely simple, this method is inherently imprecise, because it reduces all the points within a cell to the location of the BS serving it. The error produced by this method is closely related to the size of the cell, which means that it can range from tens of metres in dense urban conditions, to several kilometres and more in suburban and highway conditions respectively.

Use of additional parameters, like *Timing Advance* (TA) and received signal strength (RXLEV), in GSM networks was experimentally analysed in [14]. TA is a parameter used for synchronisation of user transmissions with corresponding TDMA (*Time-Division Multiple Access*) slots. It is a discrete-valued parameter ranging from 0 to 63, with each step corresponding to 550 metres of distance between the MS and BS, and as such can be used for ranging the MS. When omnidirectional antennas are used, the CID+TA parameters result in a circular ring centred in the BS, with a width of 550 metres, and the location estimate is, as in the case of localisation with CID only, the location of the BS. If the serving cell is a sector one, i.e. the antenna has a limited width of radiation, the location estimate is calculated in the direction of the antenna at the distance obtained by the TA parameter. With the inclusion of RXLEV parameter from up to 6 neighbouring cells, the precision of the location estimates

is increased, as this parameter is used to estimate the distance from the neighbouring cells, reducing the possible error.

Round Trip Time (RTT) measurements are used for the ranging localisation technique proposed in [15]. With the assumption of equivalent channel characteristics between each BS in range and the MS, it is possible to derive the ratio of distances between the MS and two BSs from RXLEV measurements of the signals coming from the two base stations. Such ratio defines a circle on which the MS is located, and the final location estimate is produced by finding the intersection of multiple circles, each produced by a pair of RXLEV measurements from BSs in range of the MS.

Lateration methods for localisation in GSM networks are based on measurements of ToA and TDoA and further adaptations [16]. When using TDoA measurements, the geometric-time difference is used to determine the user location. This value is calculated as the difference between the observed time difference which is measured by the MS, and the real-time difference which is the relative synchronisation difference between the two BSs used for TDoA measurement. Similar calculations are performed when ToA parameter is used in GSM, in this case with the help of additional location measurement units, which are used to calculate the offset of the MS internal clock.

Timing parameters measurements may also be taken by the network, as analysed in [17]. To avoid the problem of MS synchronisation with the rest of the network, ToA measurements are taken by different BSs, and then the TDoA is calculated. Simulation evaluation shows expected results of the suburban environment being more suitable for ranging methods, as the effects of multipath propagation have a deteriorating effect on the precision of ToA measurements in urban conditions. This results, for example, in localisation accuracy in suburban conditions with only 3 BSs being almost equal to the accuracy of localisation using 8 BSs in urban surroundings.

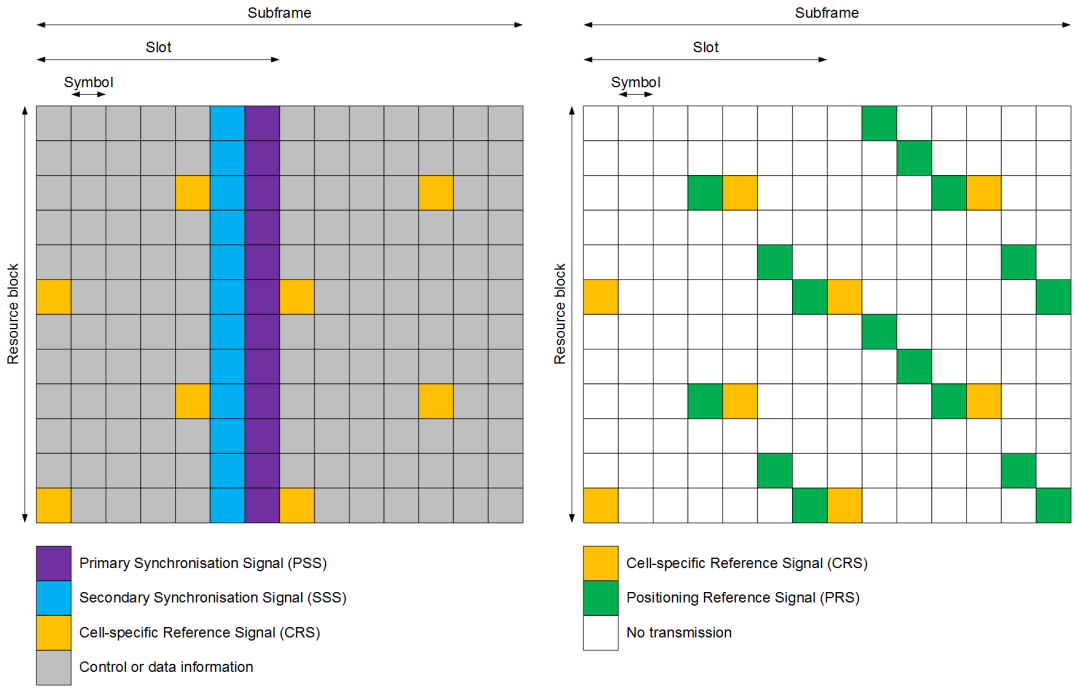
While in GSM adjacent cells use different frequencies, in UMTS all cells work on the same frequency. This causes the problem of interference in UMTS networks, which arises when MSs need to take measurements from BSs other than the one serving them. In order to mitigate such interference, time aligning of the idle downlink periods of each BS is proposed [18]. With this technique implemented, the MS is able to measure the signal from BSs further away, without interference, increasing the number of different measurements available for the localisation process, and consequently increasing the accuracy of the location estimate produced.

Different solutions using the combination of CID and RTT parameters have been proposed and implemented in UMTS networks. The accuracy of such methods depends greatly on the geometry of the BSs surrounding the MS, as well as the different sector configurations and widths of cell spacing, as this dictates with how many different BSs the MS is able to communicate [19]. Whether or not the MS is in the state of handover impacts the number of different measurements at its disposal, which results in higher localisation precision when the MS is measuring RTT associated with different CIDs, compared to the case when it is only communicating with one BS and has access to only one CID and one RTT measurement. A simple way of improving the accuracy of the CID+RTT method is to force the MS to perform handover, by changing the threshold value for adding a new BS to the active set. Two similar implementations are proposed in [20] and [21], where the threshold for adding a new BS to the active set is decreased incrementally until there are three BSs in the active set. In case the third BS is not available, the procedure is terminated when the threshold reaches a preset minimum value.

Uplink TDoA (UTDoA) is a ranging technique based on TDoA measurements performed on the signal arriving from MS to different *Location Measurement Units* (LMUs), which are synchronised and usually placed on BS sites. An implementation of this technique using *High Speed Uplink Packet Access* (HSUPA) – the data upload protocol used in UMTS networks –

is evaluated in a series of indoor field tests in [26]. The accuracy is shown to be 57 and 28 metres 67% of the time in urban and suburban environments respectively.

The development of LTE brought along solutions for the issues arising from low *Signal to Interference plus Noise Ratio* (SINR) levels of signals identifying individual cells. *Positioning Reference Signals* (PRS) as well as interference management techniques (such as low-interference subframes) have been introduced to transmissions from the BSs, with the aim of enabling a greater probability of users being able to hear the BS. PRSs are signals transmitted for the sole purpose of localisation. During the transmission of PRS, no other data is transmitted and their transmissions from different BSs are coordinated in order to avoid interference. This can be seen in Figure 2.7, where one resource block of an LTE resource grid is shown for normal and positioning subframes. The theoretical limits of localisation accuracy using these localisation-dedicated signals and interference cancellation techniques are investigated in [22]. It is shown that without coordination of positioning signals precision between 20 and 40 metres can only be achieved in cell-border areas, while the errors go up to 100 metres closer to base stations. Using interference cancellation the range of errors is limited to between 10 and 40 metres, while network coordination reduces theoretical errors below 1 metre. Allowing the PRS greater frequency width and not using this part of the spectrum for other transmission reduces the spectral efficiency of the network, but theoretically can reduce the positioning errors below 1 centimetre. Experimental performance analysis of *Observed TDoA* (OTDoA) technique in an urban scenario is presented in [23] using channel measurements. A localisation error of under 20 metres in 50% of the time and 63 metres in 95% of the time is achieved. It is shown that the FCC localisation accuracy requirements can be met using this technique, although special equipment was used for channel measurements instead of the actual UE, and perfect synchronisation of different base stations existed in the field trial.



(a) Normal subframe (b) Positioning subframe

Fig. 2.7 Resource block of an LTE resource grid

Timing measurements are susceptible to multipath propagation effects. These effects can be reduced in LTE by detecting the first arriving path, i.e. the *Line Of Sight* (LOS) path, as this is the direct, shortest path. This way, the precision of ToA and TDoA measurements can be increased. Signals other than PRS may be used for ranging purposes, but PRS is shown to have better performance in [24]. Using PRS produces lowest distance estimation errors from ToA measurements. This accuracy depends on the channel width, and may not always be enough to fulfil the standardised requirements, in which cases additional control signals need to be used in order to improve the estimate accuracy to meet the FCC requirements. A vehicle tracking algorithm using LTE ToA measurements based on *Cell-specific Reference Signal* (CRS) (shown in Figure 2.7) is presented in [27]. CRS is a signal whose transmission is mandatory, and as such is suitable for opportunistic ranging measurements. ToA estimators both with and without *Non-Line Of Sight* (NLOS) components in multipath propagation

conditions have been field tested, and localisation precision of 20 metres in 50% of the time, with full coverage, has been shown.

A collaborative approach to improving localisation in LTE networks is proposed in [25]. Apart from the standard OTDoA measurements of signals coming from the base stations, it is proposed that the users perform ranging measurements among themselves as well. As there is no synchronisation among the users, the RTT measurements are used to estimate the distance between the users. It is shown that combining the two sets of measurements is beneficial for the process of localisation, even with low SINR levels of the PRS signal. As the number of collaborating nodes increases, so does the possibility of producing a unique location estimate, without ambiguity, and the localisation accuracy.

As the accuracy of cellular system based localisation increased with the advances brought by LTE technology, so did the number of proposed solutions for indoor positioning based on LTE. Indoor localisation is a specific problem, as the alternatives in the form of various global navigation satellite systems do not exist, unlike in the case of outdoor positioning. The required level of precision also may be higher for indoor localisation, i.e. it is not just enough to estimate the location in 2D coordinates, it may be necessary to distinguish the exact room or floor of a tall building the user is located at.

An indoor positioning solution based on LTE TDoA measurements is presented in [28]. A particle filter composed of a transition model based on an indoor pedestrian movement model and a TDoA measurement model is used to calculate the likelihood of different possible states and to select the ones most consistent with the measurements to produce the final location estimate. Root mean square errors of 5 and 2 metres are shown using an experimental setup, without and with the knowledge of the initial location and the building floor plans respectively. Floor detection using LTE TDoA measurements is shown to be feasible in [29], using femtocells. A simplified model of the TDoA measurement precision in such a scenario is developed and tested with assumptions of synchronised femtocells. Vertical accuracy of

below 1 metre is shown to be possible, although these results are likely to be worse in a realistic scenario.

An indoor localisation algorithm based on a particle filter and femtocells is proposed in [30]. Uncertainty about the exact locations of the femtocells is assumed and, depending on this uncertainty the femtocells either aid (if certain) or cooperate (if uncertain) in the localisation process. Cooperation between different MSs is another part of this solution, and RTT (two-way ToA) measurements are used for this type of cooperation, while TDoA is used for measurements from the BSs. It is shown that the use of femtocells and other MSs increases localisation accuracy in indoor scenarios compared to only using macro BSs.

The errors in location estimates produced by the ranging techniques for mobile networks described above are caused by multiple factors: BS geometry and density, use of omnidirectional versus sector antennas, resolution and uncertainty of the distance estimates produced by the time measurements, multipath propagation of signals coming from BSs (particularly in urban environments) causing uncertainty of the correlation between the signal strength and the actual distance from MS to BS, as well as affecting RTT measurements. With the development of the newer generations of public cellular networks, some of the adverse effects these factors caused, like the resolution of TA measurements, have been mitigated. Some issues, on the other hand, like multipath propagation, are inherent to all wireless communication techniques, and the uncertainty caused by them cannot be fully avoided, although NLOS signal conditions have been shown to be detectable in LTE networks. More precise measurements becoming available with newer generations of communication technologies aim to reduce these uncertainties and apply advanced algorithms to detect errors and further process them.

Apart from estimating user location on their own, cellular networks can also provide assistance for GPS localisation. *Assisted-GPS* (A-GPS) is a localisation technique based on GPS, with assistance from the cellular network [31]. The assistance comes in the form

of sharing information on which satellites and at which frequencies the user may expect to receive data from, thus reducing the time needed for the location estimate to be produced. This also increases the time the MS can spend listening to GPS signals, allowing it to hear lower strength signals than it would normally be able to.

Use of Cell-ID parameter for conserving energy the amount of energy GPS consumes is proposed in [32]. This localisation algorithm aims to limit the usage of energy-hungry GPS, and use the information about the CIDs through which the user moves over time to determine its location. A database of the CID readings along routes taken by the user, with occasional GPS readings, is maintained. The database is later used to compare the sequence of CID readings without GPS, and, if significant matching is discovered, compute the user location with precision greater than the one offered by the basic CID technique. Such an approach exploits the patterns user mobility exhibits, both in spatial and temporal domains, which may limit its applicability in scenarios where new routes are being used, or when the user is not on the move constantly.

Pattern matching methods in cellular networks

Another big family of solutions for positioning in cellular networks are pattern matching solutions which can be divided into ones modelling signal propagation using analytical propagation models [33, 34] and fingerprinting ones, although these two approaches can complement each other, as in [33].

A one-to-one mapping between the MS location and the vector of received signal strength measurements is assumed to exist in [33] and the performance of localisation based on this assumption is evaluated. This work shows the limits of such an approach, depending on measurement uncertainty, BS geometry and the number of available BSs in range. For a cell with a 1 kilometre radius, the simulated localisation accuracy in 67% of the time is shown to

be 250 metres. The standard method of collecting measurements in the field and creating a database to compare the new measurement to is proposed for practical implementation.

A localisation solution for users with measurement information only available from the BS they are currently connected to is proposed in [34]. A Bayesian method is used to calculate the density of user location based on which BS is currently serving the user, the known locations of this and surrounding BSs, and a signal propagation model. To further improve the precision of this method, additional information is built into their model, like RTT measurement value and sector antenna directions. IT is shown that an improvement of 20% for the mean error is achieved compared to just placing the user uniformly in an arc calculated based on RTT and BS sector information.

Fingerprinting methods, like the ones proposed for outdoor [35] and indoor [36], use in cellular networks, assume the existence of a database of measurements taken before the deployment of the localisation system, during the offline phase, which presents the main drawback of such approach. This workload can be reduced by dividing the area of interest in a grid, thus reducing the number of needed measurements at the expense of precision [37, 38], or by applying machine learning techniques [39] to learn about the propagation parameters in the given environment. Accuracy of pattern matching can also be increased using cell-ID and RTT measurements [40–44], adding WLAN networks RSS measurements to the fingerprints [45] or device-to-device cooperation [46]. Accuracy of the propagation model-based approach depends on how well the analytical or empirical model used is able to describe the environment, as well as how diverse the environment itself is, while the accuracy of the solutions using fingerprint databases depends on the number of measurements, especially when the variance of RSS measurements in a single location (or grid cell) is high.

A database correlation method for GSM networks is proposed in [35]. Storing of signal or channel properties relative to individual locations in a database is done with the goal of using this information to compare it to any new measurements performed by the user

(or the network for the user) needing a location estimate. Apart from acquiring the actual measurements, which may be a difficult task, a possibility of using mathematical models of signal propagation for the areas of interest is also proposed. The accuracy of this method is shown to be 44 and 74 metres in urban and suburban environments respectively.

Use of GSM fingerprinting for indoor localisation is analysed in [36]. Algorithms relying on single cell measurements, 6 different cells, and 35 different GSM channels are compared. It is shown that by applying a greedy feature selection algorithm to remove the cells whose signals are either too stable across different locations or are too noisy, i.e. varying over time, can significantly increase the localisation precision. K nearest neighbour method is used to compute the user location as a centroid of the locations of the K closest training points from the database.

In order to avoid having to perform a large number of measurements from many different locations in the area of interest to populate the fingerprint database, some solutions apply the gridding approach (shown in Figure 2.8), as in [37]. The whole area of interest is divided into grid cells, the size of which represents the trade-off between accuracy and complexity. All the measurements taken inside one cell are put together, and a histogram of RSS measurements is created for each BS for each cell. The centroid of all measurement points inside a cell is used as the location estimate for that cell. The algorithm is able to provide a more precise estimate by analysing each individual fingerprint from the selected cell, using nearest neighbours method. Field tests have shown this method to achieve a median error of 42 and 28 metres in rural and urban scenarios respectively.

Another solution employing gridding approach as units for fingerprint matching is proposed in [38]. This solution addresses 3D localisation by creating a separate grid for each floor of a building. On top of the standard pattern matching between user measurement and the collected database, this algorithm introduces correlation tests for points surrounding the

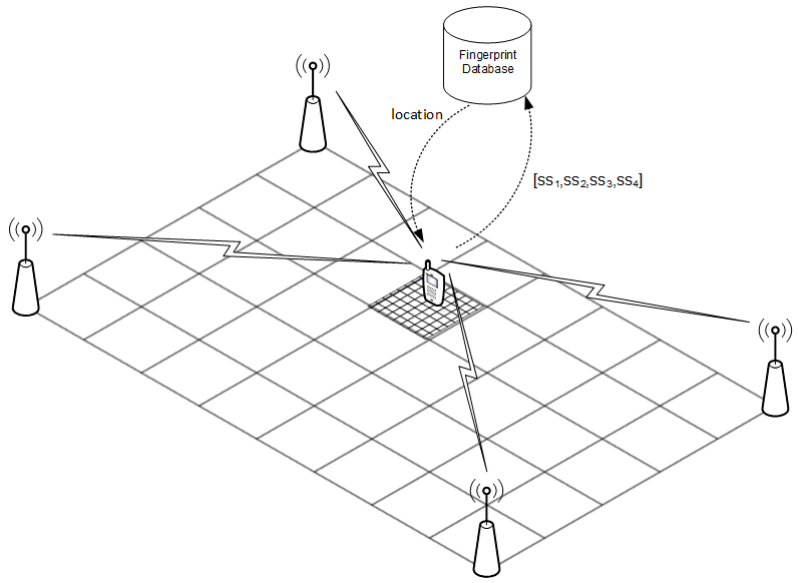


Fig. 2.8 Illustration of the gridding approach used in fingerprint localisation

chosen estimate to ensure the validity of the estimate. This addition is reported to achieve a localisation error of 21 metres in 67% of the time.

Avoiding the initial effort of compiling a database of fingerprints associated with known locations can be done using machine learning, as in [39]. Both semi-supervised and unsupervised learning methods are proposed. The supervised learning method is making use of a small amount of labelled training data and spatial interpolation to extend the "knowledge" about location across unlabelled fingerprints, while the unsupervised one makes use of propagation models to estimate expected signal measurements at different locations. The semi-supervised method is shown to be more accurate than methods relying on labelled data only when using the same amount of training data, while the unsupervised model based on a well-fit propagation model performs even better.

Fingerprinting based on the cell ID parameter is proposed in [40]. Each CID value is associated with a polygon region in which the user is located based on the given CID. This approach is illustrated in Figure 2.9. A polygon contraction algorithm which provides a balance between accuracy, i.e. keeping the polygon as small as possible, and confidence, i.e. keeping the polygon big enough for a certain level of confidence that a CID reading

corresponds to the correct polygon. Additionally, handover regions may be used for defining additional polygons, thus increasing accuracy. Using simulation, the algorithm is shown to be able to produce location estimate the size of a quarter to a half of the cell size with high confidence, which is an improvement to the basic CID technique. An improvement to this algorithm is presented in [41], using RTT measurements. The RTT measurement provides a circular strip around the BS, in which the user is located. The intersection of the cell polygon and the circular strip provides a more precise estimate of the user location, even with information from only one BS available for positioning. With measurements from multiple BSs at disposal, this method becomes an adaptation of triangulation, where each BS produces an arc, rather than a full circle. In experimental field tests, an accuracy of 78 metres in 67% of the time, availability of 95%, and positioning time of under 0.2 seconds in 65% of the time have been shown. A further improvement [42], dealing with contraction of narrow polygons created as the intersection of the original cell polygon and zones assigned to RTT and TA measurements, was shown to be helpful particularly in LTE scenarios with unfavourable geometry of BSs. An addition to this algorithm, enabling it to produce three-dimensional location estimates is presented in [43]. Polygon corners with two-dimensional coordinates are added an altitude dimension based on additional estimates from any lateration technique (OTDoA, UTDoA), and shape conversion is performed on the polygon with altitude into a 3D point with ellipsoidal uncertainty.

An energy efficient CID and RSS pattern matching algorithm enhanced with sensor information for tracking mobile users on a map is proposed in [44]. Fingerprints containing cell ID and signal strength are first assigned to grid cells and the user movement over time is then smoothed. The output obtained that way is then analysed together with phone sensor data (if available) which may indicate the speed of movement, or the user making a turn, to best match the sequence of location cells to particular road segments on a map.

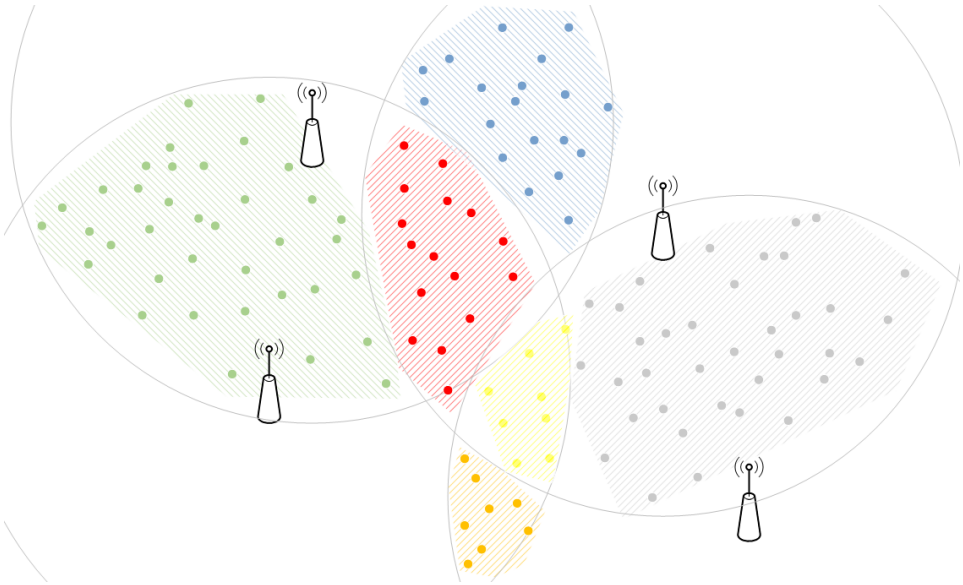


Fig. 2.9 Illustration of the cell polygon regions

Fingerprinting methods can combine measurement data from different network technologies, like the LTE and WLAN combination examined in [45]. Including the additional WLAN network measurements to the LTE network measurements is experimentally shown to increase the accuracy compared to using just LTE measurements. The 67% accuracy for the method with combined data is 16 metres using a 20 metre grid, and 25 metres using 40 metre grid.

Device-to-device communication is emerging as a standard part of the latest releases of LTE network standards and is being exploited as a tool for improving pattern matching based localisation. In [46], fingerprint matching is used only to find candidate points for the user location, but the final estimate is computed based on cooperation between users, which perform RTT measurements. These measurements are used to choose the most likely candidate point as the estimate. This cooperative process is shown to improve the accuracy of location estimates, which grows with every additional user participating in RTT measurements.

The lower implementation complexity of pattern matching localisation solutions, in comparison to range-based solutions, comes at the cost of lower accuracy. As can be

concluded from the works mentioned above, the fingerprinting positioning error is being reduced with the development of new mobile networks and localisation algorithms, which are using parameters other than just signal strength, as well as concepts such as user device cooperation. This leaves efficiency of the offline data collection process as the main issue of this method, which is a problem that is currently being addressed mostly by using propagation models, to either fill the gaps in the training data or create the whole training dataset, thus completely removing the need for an offline phase.

Wireless Local Area Networks

The architecture of WLAN networks is similar to that of cellular networks, in the sense that there exist fixed access points which provide network access to users in range, and through which all of the communication flows. On the other hand, the fact that WLANs are mostly deployed in indoor environments increases the chances of multipath signal propagation, the range and coverage of the individual APs are significantly lower than those of the BSs, which means that accurate range-based localisation is difficult to perform in WLANs. This, along with the fact that the number of available APs in range can easily be much greater than the number of BSs, increasing the number of reference points for fingerprinting methods compared to cellular networks, are the main reasons why most of the localisation solutions for WLANs are based on range-free, pattern matching algorithms.

Range-based localisation in Wireless Local Area Networks

Among the few localisation methods that are not based on pattern matching algorithms, there are solutions which try to estimate distance between APs and users based on timing measurements [47], fitting the set of RSS measurements into a system of physical constraints [48], or more indirectly, through propagation models [49, 50], or by defining different contours around APs which correspond to different RSS measurement values [51].

Performing timing measurements in early WLAN networks is proposed in [47], where *Direct Sequence Spread Spectrum* (DSSS) and *Orthogonal Frequency Division Multiplexing* (OFDM) standards for wireless networks are examined. Synchronisation between AP and user is instrumental for successful timing measurement, and depending on the type of packets used for these measurements, different errors in distance estimates can be expected, ranging from tens to hundreds of metres. Multipath propagation and NLOS conditions additionally degrade the ranging performance.

An algorithm which analyses the RSS measurements between users and different APs is proposed in [48]. Assuming all of these measurements are constrained by the physics of signal propagation, a genetic algorithm is used to estimate the positions of APs, their transmit power and the path loss exponents, from the set of RSS measurements. This solution only requires occasional ground truth calibration, which is obtained by means of GPS, as the algorithm itself solves the relative locations of APs and users, and need a reference to produce location estimates in global coordinates. The advantage of such an approach is that the actual AP locations do not need to be known, which distinguishes this solution from other RSS-based solutions.

Modelling signal propagation in indoor settings is considerably more difficult than doing so in outdoor conditions, which is why the authors of [49] propose using a dynamic path loss model for each AP, which learns from the data collected from RSS measurements performed at known locations. An illustration of RSS values for a propagation model based on measurements is shown in figure 2.10, where the darkness of the colour signifies the strength of the signal, and the dots represent experimental measurements. It is shown that taking the propagation parameters uncertainty into account and using a probabilistic approach results in higher accuracy than assuming these parameters are known and fixed. Bayesian-based path loss parameter calibration method for localisation using RSS measurements is presented in [50] as well. A particle filter is used along a state transition model to estimate

the user location. The model learns continuously and shows a decrease in location estimate error over time. Again, it is shown that better localisation accuracy is achieved by allowing for the path loss parameters to be dynamically estimated, rather than assuming them to have fixed values.

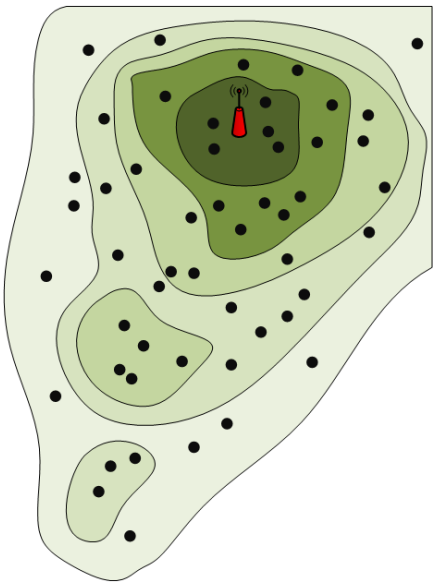


Fig. 2.10 Illustration of a propagation model

A combination of lateration and fingerprinting is proposed in [51]. Instead of translating RSS measurements into distance estimates, this system uses reference points. For each RSS measurement from a single AP, a contour of reference points with the same RSS value is produced. These contours are used the same way circles are in classic trilateration technique. Device calibration is done using signal correlation, to accommodate for the use of heterogeneous devices, without the need for offline calibration. In order to fight multipath propagation, this algorithm assigns higher weights to contours from APs with higher measured signal strength.

Pattern matching localisation in Wireless Local Area Networks

The body of work concerning fingerprinting localisation in WLANs is vast, especially compared to the one relating to lateration, which, as can be concluded from the works listed above, is mainly based on propagation models, rather than precise timing or signal strength measurements for distance estimation. As the fingerprinting solutions used in cellular networks, the fingerprinting solutions for WLANs are usually assuming an offline data collection phase. In general, the approach to forming the fingerprint database is similar to the one described in [52]. To accommodate for the dynamics of the environments, some solutions propose creating multiple databases [53] or collecting RSS measurements over time to form histograms corresponding to each reference point instead of exact, unique values [54, 55]. While these approaches result in an increase of measurements to be obtained in the offline phase, there exist solutions which propose the use of probabilistic signal propagation models [56, 57], to reduce the effort for database creation. The accuracy of fingerprinting methods can be increased, but this increase usually comes at the expense of the increased system complexity as well. Location tracking and prediction [58–60] assumes additional system elements for tracking and prediction to be implemented, while reducing the effects of noise on the RSS measurements can be achieved by either additional distance estimation being performed [61] which assumes either very precise RSS measurements or additional ranging hardware, or by using multiple interfaces on a single device [62], which is in itself a more complex approach.

One of the earliest fingerprinting solutions for WLANs was proposed in [52]. The offline phase considers performing measurements in reference points throughout the area of interest. As in any standard fingerprinting procedure, the offline measurements are used to find the closest (one or more) reference points based on the unknown location measurements. As an alternative to collecting online data, the use of a radio propagation model is presented. This method sacrifices localisation accuracy but simplifies the setup procedure significantly. An

enhancement to this system is presented in [53], where continuous user tracking is used for disambiguation of potential user locations. In order to account for various environmental conditions, multiple radiomaps (sets of fingerprints) have been created. In order to calibrate the system for the correct radiomap to be used, a simple test is performed. As each AP is at a known location, by estimating its location as if it were a user in an unknown location, using signal strength from all other APs, the model which gives highest accuracy is determined.

Fingerprinting methods may also function on a probabilistic basis, like the one described in [54] and [55], where RSS measurements are observed over time at reference point locations, and histograms of these measurements are stored. These histograms are then used to characterise the individual AP signal strength throughout the area covered by the reference point. When a user with an unknown location performs RSS measurements, its location is estimated in the form of a probability distribution, according to the a priori knowledge obtained during the reference point measurements. This solution also introduces location clustering in order to reduce the complexity of calculating the location estimate probability distribution. Locations which are covered by a certain number of common APs are placed in the same cluster, thus reducing the number of possible location candidates from the whole set of reference points to only the subset of points which share the same APs with the measurement performed at an unknown location.

As previously discussed, one of the main drawbacks of the fingerprinting method is the amount of effort needed for collecting reference data in the offline phase. One way of reducing the workload of the offline phase is by using a propagation model, as in [56]. The algorithm proposed in this paper is using a small number of actual reference measurements to train a nonparametric model to produce the full fingerprint database. An experimental setup was used for evaluation, and it was shown that localisation using the proposed method outperforms the traditional fingerprinting algorithm using actual measurements to create the radiomap. Furthermore, the proposed method is able to maintain similar levels of accuracy

with only 5% of the original training data. It was noted that some APs broadcast more than one *Medium Access Control* (MAC) addresses, and that aggregating these separate signal measurements can help in cases when only partial training measurements are available. A similar approach is proposed in [57]. A clustering method is used to divide the whole localisation area into sub-regions based on sparse measurements and the proposed regional propagation model is used to reconstruct the database of the whole area.

A probabilistic location tracking algorithm using a nonparametric information filter is presented in [58]. This solution aims to improve the accuracy of fingerprint radiomap-based location estimates by predicting the future user location using the filter. Once an expected location is computed, the algorithm can decide which APs are relevant to the localisation process, as well as which reference points are part of the possible region of interest, in which the user is assumed to be located based on the prediction. Predicted location is also used to detect possible RSS measurement outliers, which are discarded in order to prevent the error to be propagated through the prediction mechanism. Applying restrictions on the output of the localisation process based on previous location estimates in also part of the algorithm proposed in [59]. A similar method of creating a region of interest in which the location estimate is allowed to be placed is used. This solution utilises Kalman filter to reduce localisation errors, assuming linear movement of the user with the unknown location. In order to accommodate for the errors Kalman filter produces when users make turns, it is necessary to have map data of the area in which the system operates. In [60], the optimal user location estimate is computed using particle swarm optimisation technique. This localisation algorithm also uses Kalman filter to process the estimates and reduce the errors by tracking the user location over time.

Distance bounds are fused with the standard fingerprinting procedure in [61] to reduce the effects of noisy RSS measurements. The distance information may be spatial, i.e. as a result of ranging using direct communication with some anchor nodes in the surroundings, or

temporal, i.e. the target itself estimates the distance it has traversed over a certain period of time, by means of internal navigation system available in many modern devices. Using this additional distance information, the expected signal difference between the unknown user location and all reference points is computed. Given the set of these differences, optimal values of weights determining the contribution of each reference point to the final location estimate are estimated.

Mitigating the effects of noise and fading on RSS measurements by using multiple wireless interfaces is proposed in [62]. The authors test a different number of multiple interfaces on the same user device, using different positions and orientations. A low correlation between measurements performed at the same time and location by different interfaces with the same hardware is observed, which suggests a high impact of noise and fading to these measurements. Different interface measurements are combined in one in order to acquire RSS values which exhibit greater stability over time. Several methods for combining individual measurements are tested. A clear improvement in localisation accuracy is shown as the number of interfaces increases.

Fingerprinting methods are susceptible to issues induced by device heterogeneity, as different wireless interfaces on different devices are likely to report varying values of the RSS measurement under the same conditions. This can easily result in reference points far from the true location of the user whose location is estimated being recognised as nearest ones to the measurements obtained by the user device. Furthermore, because of this variance, reference points from completely different regions of the area in which the localisation system operates may appear as equally valid candidates. A number of calibration solutions have been proposed, to adjust radiomaps in a way that location estimate does not depend on the type of hardware used by the user being localised. In [63] a linear approximation is used to map RSS measurement performed by any hardware into one obtained by the hardware used during the offline phase which is the reference measurement contained in

the radiomap. Supervised calibration, in manual and quasi-automatic variants, as well as unsupervised automatic calibration, are proposed. Manual calibration assumes assigning ground truth values of user location to a certain number of measurements, while quasi-automatic calibration is done by applying different values for linear coefficients, making use of the fact that wrong values produce RSS measurements which do not match any of the reference points. In an attempt to completely remove the user input from the calibration procedure, automatic calibration is running simultaneously with localisation and tracking of the user but is reported to perform worse than the supervised calibration variants. A solution for RSS variance based on devices calibrating themselves during the online phase is presented in [64]. This solution uses histograms of the reference and the RSS measurements made by the user device. These probability distributions are used to compute the linear mapping coefficients which allow the measurements performed on the user device to be translated into the one obtained by the reference device. The localisation results using this calibration method are much closer to the ones obtained using a device-specific radiomap, showing significant improvement compared to non-calibrated results.

The variance of RSS measurements is caused not only by the user device differences but also by the placement of the device, the direction of movement and temporal environmental changes, as shown in [65]. In order to solve the variance problem, a tracking system based on RSS peak detection is proposed. This approach avoids the need for calibration, by assuming that regardless of the differences in RSS measurements performed by different devices, in different positions or directions and at different times, the peak value of the received signal is preserved around the same location. As the peaks in RSS measurements appear sporadically, the availability of this localisation system over time is maintained by including the internal navigation system (accelerometer and digital compass) information to the algorithm.

The authors of [66] investigated the behaviour of different wireless chipsets and concluded that even devices from the same vendor can report significantly different RSS readings.

In addition, even measurements from a single device were not consistent over time. This type of issue can reduce the applicability of fingerprinting localisation methods, by rendering the radiomap information measured in the offline phase useless in many online use scenarios. Some devices exhibited temporal patterns, which suggests that applying filters may increase the suitability of measurements, while others produced measurements which were independent of the distance to the APs. Interference is pointed out as one of the possible causes for inconsistent measurements, as the less occupied 5 GHz band was found to be more suitable for RSS measurements than the 2.4 GHz one.

Performance of fingerprinting algorithms is influenced by the metric used for comparing the fingerprints stored in the radiomap to the ones obtained by the users with an unknown location. In [67], a comprehensive analysis of different ways of measuring the distance between fingerprints in RSS space is presented. The Euclidean distance, which is most often used, may not be appropriate, given the possibility of measurements from certain APs may be missing in some RSS vectors. Authors point out the common practice of treating the signal level measurements expressed in decibels as if they are linear, rather than logarithmic values. It is demonstrated that using Sørensen distance and an alternative RSS value representation achieves better localisation results compared to the standard Euclidean distance and linear RSS representation combination. Additionally, the practice of removing weak AP measurements was shown not to benefit the performance of localisation.

Fingerprinting methods for localisation in WLANs listed in the brief overview presented above aim to improve the accuracy of location estimates. While they achieve that to a lesser or greater extent, some large location estimate errors cannot be totally avoided, as noted in [68]. The presence of noise and fading, the density of reference points and APs, RSS variance, device heterogeneity and dynamic environment are all recognised as possible sources of localisation errors. Furthermore, the way decibel measurements are treated is listed as a source of error (as earlier noted in [67]), as well as the uncertainty introduced by

quantisation of RSS values. It is concluded that existing fingerprint methods mostly operate well in areas close to APs, in comparison to peripheral regions of the localisation area.

2.4.2 Infrastructureless networks

The main reason localisation of nodes in infrastructureless networks is a challenging task is the lack of a centralised infrastructure of fixed communication points such as APs or BSs, which results in a supporting system of localisation references being nonexistent. The nodes in such networks are usually not assumed to be equipped with the technology capable of performing accurate ranging measurements, as following the spirit of easy deployment and low maintenance that ad hoc networks are designed in, the nodes are usually simple and low cost. Furthermore, if node mobility is assumed, this challenge only becomes more difficult as the topology of the network is highly dynamic with the nodes coming in and out of each other ranges making all measurable relations between them very unstable.

Many localisation solutions have been proposed which are based on using dedicated localisation nodes, called beacons or anchor, the location of which is known. Such approach to localisation results in some solutions which are in principle very similar to the ones existing in infrastructure-based networks, particularly ones employing lateration based on direct distance estimates from timing or RSS measurements or fingerprinting approaches, as the beacon nodes represent a sort of localisation infrastructure. Having this in mind, the literature review presented below will focus on distributed range-based and connectivity-based localisation solutions in ad hoc networks. Infrastructureless wireless networks have found one of the main uses in sensing applications, which is why most of the localisation solutions are proposed for *Wireless Sensor Networks* (WSNs), but are mainly applicable to any ad hoc network in general.

Localisation based on single hop connectivity

Nodes in ad hoc networks can determine their location based on the information about which anchor they can receive signals from, i.e. inside the range of which anchor they are located. Different implementations of this approach are proposed in [69–72]. The precision and accuracy of such solutions are highly dependent on the number, distribution and the transmission range of the anchor nodes, as well as the environment in which they operate.

Localisation based on proximity to anchor nodes is proposed in [69]. Each anchor emits beacon packets at a known rate and the nodes in the network are listening for these packets. Knowing the rate of beacons being sent, and the number of beacons received from an anchor over a period of time, each node can determine the ratio of received beacons. Based on a preset threshold of received ratio of beacons, the node determines whether it is located within an anchor's range or not. The node then estimates its location as the centroid of the locations of the anchors within whose range it is located in. Resulting accuracy of the location estimate greatly depends on the ratio of the anchor density and their range. Furthermore, the anchors are assumed to be located in a deterministic way, on the vertices of a square grid. A more general solution based on the same idea, allowing non-uniform anchor placement, is proposed in [70]. An improvement to localisation accuracy is achieved by computing a rectangular area bounding the intersection region of the anchor ranges within which the node is located, as shown in Figure 2.11. Instead of finding the centroid of anchor node locations, a centroid of this bounding box is now used as the final location estimate.

Another geometrical connectivity-based approach is presented in [71]. Instead of using anchor ranges, the nodes run the point-in-triangle test, trying to determine whether they are located inside or outside a triangle formed by three anchors. I order to perform this test, nodes exchange RSS measurements from anchors with their neighbours, to determine if any of their neighbours are simultaneously closer to all three of the selected anchors. In case there is no neighbour satisfying the condition above, the node concludes it is located inside

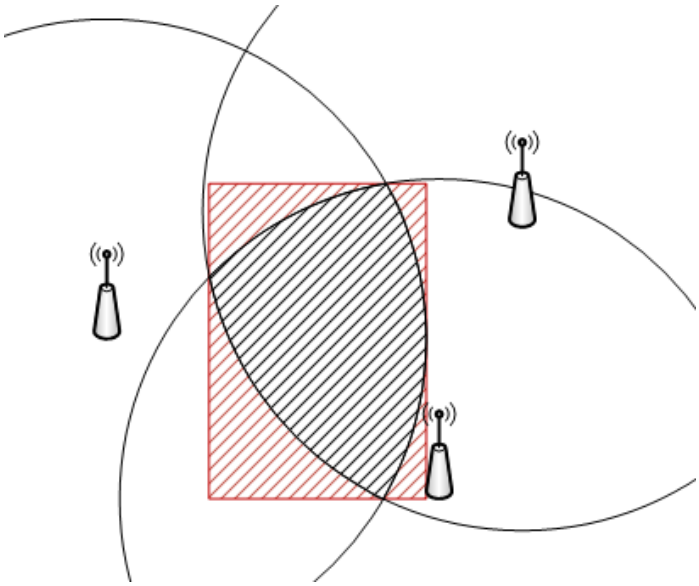


Fig. 2.11 Illustration of an estimative rectangle of ranges intersection

the triangle. This approach assumes anchors with much greater range than regular nodes, in order for them to cover as much of the localisation area as possible, so the nodes can have a greater number of three-anchor combinations forming triangles. A square grid over the whole localisation area is formed, and each square is given a score, which is initially zero. The score is increased for every positive constraint covering it, i.e. a triangle the node has estimated it is within, and a decreased for each negative constraint, i.e. one which the node has estimated it is outside of. The final location estimate is calculated as the centroid of the grid squares with the highest score.

The solution presented in [72] introduces virtual force refinement of the location estimate derived by the positive connectivity constraints. Anchor ranges are used to determine portions of space in which the node may be located. Again, estimative rectangles are used to represent these portions of space. Instead of creating a square grid covering the whole area of the network, this algorithm only considers a square grid within the estimative rectangle and estimates the location of the node as the centroid of the squares which have the highest score, similarly to the previously listed solution. The negative constraints are observed as virtual forces, by analysing the locations of anchors whose range the node is not within, relative

to the location of the anchors which the node can hear. These virtual forces determine how much and in which direction the location estimate of the node should be corrected. The information about the anchors which the node cannot hear is obtained from neighbouring nodes, through multi-hop communication.

Localisation based on hop-count

While the solutions described above rely on single hop connectivity information, there is a family of solutions, based on the approach of [73], which consider multi-hop connectivity information and use hop-count to estimate distances between nodes. The distances individual communication hops cover can have high variance, leading to inaccurate location estimates, which is a problem tackled by hop-count analysis [74–76] or by using RSS measurements-based ranging in addition to the hop-count metric [77–79].

In the algorithm proposed in [73] (DV-hop), the anchor nodes with known location propagate messages throughout the network, using multi-hop communication. Once an anchor receives a message from another anchor, it can determine the number of hops the message has traversed, and by knowing the other anchors and its own location, calculate the average length of a communication hop. On the other hand, regular nodes maintain their routing tables, containing hop counts to different anchors. The length of a hop is propagated from each anchor, enabling nodes to estimate their distance to anchors, and ultimately, perform triangulation to estimate their location. In addition, the authors propose neighbouring nodes measuring the distance to their neighbours using RSS. This method has a higher resolution than simple hop count but is more sensitive to signal strength uncertainties. Finally, a pure triangulation approach is proposed by nodes estimating their location based on distance estimates to neighbouring nodes which have already estimated their locations.

A number of improvements to the DV-hop algorithm have been proposed. The one in [74] deals with hop count to distance mapping by deriving the expected hop progress, the distance

in the direction of the anchor node that an average hop a packet traverses on the path from a node towards the anchor. It is shown that the expected hop progress is determined by network parameters such as node density and transmission range and that it is possible to directly calculate it from expected node connectivity. As the messages from anchors propagate the network, the hop count is incremented in a usual way, and the hop progress is updated by each node based on its own number of neighbours. A simulation-based evaluation shows that using expected hop progress to estimate the distance to anchors improves the localisation accuracy compared to the basic setup. Scaling the hop distance based in an optimal way with respect to all anchors in the network is proposed in [75]. The algorithm works based on propagating hop count values between all anchors throughout the whole network, so each node can be aware of all the distances and hop counts between the anchor nodes, and is able to compute the right scaling factor which will be used to map hop counts into distances to anchor nodes. Two different adaptations to the mapping of hop counts into distances in heterogeneous networks, i.e. networks where nodes' communication range is not uniform, is proposed in [76].

The idea of using hop count to estimate the distance between nodes is also used in [77], where a two-step localisation algorithm is presented. In the first step, similarly to DV-hop the anchors broadcast their location and the messages are propagated to all nodes, enabling them to convert the hop count into distance using the average hop distance. In the second, refinement stage, the nodes estimate the distances to their neighbouring nodes and perform triangulation. The location obtained in that way is assigned a confidence level based on the number and topology of the immediate neighbours, as well as their location confidences. Individual node's location confidence level determines the weight with which constraints involving this node will be regarded by the neighbouring nodes when performing triangulation. Hop counts conversion to distance from the anchor by finding the average of neighbourhood hop count to the anchor is proposed in [78]. The authors also propose the

use of quantisation of RSS measurements to improve the resolution of the distance estimates compared to the one produced by hop count conversion. In [79], the authors propose adding RSS measurements as a method of improving the distance estimates. Distance estimates are first produced by an adaptation of the original hop count method, referred to as virtual hop count, which accounts for the fact that the distance corresponding to individual hops should not be a fixed value. Then, RSS measurement-based distance estimates are generated and compared to the virtual hop ones. Nodes are marked as good or bad, based on how well the two sets of distance estimates match, and this assigned quality is used to weight an individual node's contribution to localisation calculations.

Localisation based on network-wide information

The approach of fitting estimated distances or hop-counts between all pairs of neighbours into a topology which would satisfy these proximity constraints is popular for localisation in ad hoc networks. One of the few solutions which do not assume the existence of anchor nodes is [80], which relies on nodes individually forming local coordinate systems and patching them together as the algorithm advances. While this approach is sensitive to the precision of distance estimates between single hop neighbours, in [81] a number of anchors are introduced to help reduce the errors arising from inaccurate distance estimates. Multidimensional scaling [82, 83], semidefinite programming [84, 85] and other optimisation techniques have been proposed [86–88] to improve the process of solving the system of proximity constraints. The main drawback of the solutions from this category is that they assume full knowledge of all single hop distance estimates or connectivity information, which limits them to networks of static nodes. During the time needed to propagate this information across the whole network, if node mobility is introduced, this information may become outdated. Solutions from the DV-hop family can, in general, suffer from the same issue, although hop distance estimates can be updated much more dynamically than full network topologies.

An algorithm based on distance measurements between neighbouring nodes is presented in [80], where each node starts by building its own coordinate system based on choosing two neighbours and deriving their relative positions using the cosine law to compute the angles of the triangle it forms with them. The same process of using triangle properties is carried on to determine the locations of all of the two hop neighbours it has information about. This results in each node having a local coordinate system which it is a centre of and has an arbitrary direction. Using the relative coordinates of nodes present in two different nodes' coordinate systems, these two nodes can align the directions of their relative coordinate systems by rotating or reflecting them, after which it is possible to merge these two locally centred coordinate systems into one. Repeating this procedure among nodes leads to the formation of a universal coordinate system, in which each node can compute its own and other nodes' locations using one node as the universal reference. Error mitigation is performed by the residual weighing algorithm, which reduces the difference between the estimated location of the node and the one conforming to the constraints of the measured distances by its neighbours.

A collaborative multilateration localisation algorithm is proposed in [81]. Similarly to the solution in [80], this algorithm has nodes determining the distance between themselves but introduces a portion of anchor nodes which have accurate information about their location, which helps reduce errors and their propagation as the nodes exchange location and distance estimates. The algorithm starts by forming of collaborative subtrees, i.e. configurations of nodes or anchors which, given their neighbourhood, have a uniquely possible location solution. Only these nodes are allowed to participate in the second phase of obtaining initial location estimates based on estimated distances from ranging measurements between neighbouring nodes. The final location estimates, both for the nodes participating in the second phase and all other nodes, are produced by minimising the difference between

measured distances and the ones computed from initially estimated locations. Kalman filter is used to update the location estimates using the measured distances.

Multidimensional scaling is a technique often used for node localisation because of its ability to compute spatial structures in the data it is presented with. Two similar algorithms, one based on connectivity (hop count) [82], and the other [83] based on distance estimates, use this technique estimate node locations. The two algorithms follow roughly the same steps. First, random locations are assigned to all nodes. Then the difference between the distances of assumed node locations and the estimated ones are reduced. The set of relative location estimates of all nodes which are produced as a result of this step later needs to be aligned with absolute locations using certain nodes whose positions are known. Location estimate alignment may involve shifting, rotating or reflection of coordinates and at least three nodes with known location are needed for this procedure. The algorithm proposed in [83] can also work for individual node localisation, rather than working on the set of all nodes at once. The node requiring localisation estimates relative locations of its one-hop neighbours, and this one-hop neighbourhood procedure is repeated until three anchor nodes are reached, after which the starting node is able to align its relative location to an absolute one. During this procedure, the nodes along the paths to anchors are also able to acquire their absolute location estimates.

Semidefinite programming is another technique which can be used to compute node locations based on distance estimates and known anchor locations, as shown in [84]. The algorithm works by converting the quadratic distance constraints into linear ones. Again, the function which is minimised is the one representing the sum of errors between the estimated locations and the constraints imposed by distance estimates between nodes. An improvement to this method is proposed in [85], where a gradient search method is used to compute the location estimates more accurately. This is done by moving the location estimate of each node in the direction opposite to the gradient direction of the sum of error square function

after the initial estimates have been produced by the original method. It is shown that the gradient search method cannot work on its own, as in the case of just placing each node's estimated location in the origin of the coordinate system, the final outcome of this method produces much greater errors in location estimates. This is because the gradient search is only able to converge towards the local minimum. A technique similar to gradient search, called simulated annealing is proposed in [86]. This technique, unlike the gradient search, can take the convergence outside of the local minimum, by temporarily increasing the error, in order to reach the global minimum. Having the ability to do so, makes this technique a viable standalone solution for location estimate computing. As in other solutions based on estimating locations based on distance estimates between nodes, the problem of ambiguous node locations may arise, that is, a node or a group of nodes may be located in two different locations, both of which satisfy the distance constraints. In [87] a solution to this problem is proposed. If the node is placed in the wrong neighbourhood, i.e. is close to nodes to which it has no direct distance measurement to a refinement procedure is initiated. The minimum error of the ambiguous location estimate is defined and included in the new cost function which is then minimised to solve the location ambiguity. Minimisation of the variance in one hop neighbour distance in connectivity based localisation is proposed in [88], where initial estimates obtained using the standard hop count method are refined so that each node is pushed towards the location in which it would have uniform distances to its neighbours.

Another topic in ad hoc networks localisation is network deployment in irregular areas. Such areas cause the issue of holes appearing in topology which distort the relationship between the hop-count metric and the distance between two nodes. To solve this problem, partitioning of the network [89], analysing the deviation of the shortest path between nodes around the hole [90, 91] and reliable anchor detection [92–95] have been proposed.

An improvement to the algorithm from [82], enabling it to deal with irregularly shaped regions where the physical topology of nodes exhibits "holes" is presented in [89]. In order

to do so, the improved algorithm is decentralised unlike the original, centralised one, and is able to put together patches of relative maps which have been created independently based on the number of common nodes in them. This algorithm is able to solve the problem of the discrepancy between the Euclidean distance and hop count for certain node, which is common in the conditions described above, when physically near nodes have a high hop count in the shortest network path between them. Another method for mitigating the negative effects the presence of holes in node topology produces is presented in [90]. This algorithm works with only three anchor nodes, and it does not assume the creation of partial maps and their stitching together. It works by detecting the holes in topology and creating virtual holes around the nodes which are located at hole edges. This manipulation allows the algorithm to calculate the angles at which the shortest path between two nodes is bent in order to go around the hole and accurately estimate the real difference between two nodes based on this information and the standard hop count approach. Anisotropic topology detection is also examined in [91]. Reliable anchors are detected by the degree of hop count deviation to the expected one, given the known anchor locations, and reliable anchor pairs are used for location estimates. The same problem is approached in [92], where anchors are classified based on the hop count patterns packets arriving from them exhibit. Once an anchor is classified, different distance estimation technique is used. The work presented in [93] deals with both nonuniform node distribution as well as the irregularity of the region in which the nodes are located, by determining reliable anchors. In [94] a similar algorithm dealing with sparse anchor distribution is proposed. A Pascal triangle solution for localisation based on average neighbour distance, hop progress and angle at which the shortest path between two nodes bends in anisotropic regions can be found in [95].

Infrastructureless networks localisation summary

Depending on the type of approach, the accuracy of localisation algorithms designed for ad hoc networks may be influenced by a number of factors. The solutions relying on anchor nodes are mostly dependent on the number, density and location of these nodes, especially in irregularly shaped network topologies. As in many cases in cellular networks positioning, higher accuracy is much easier to achieve closer to the reference points. Nodes with a high number of neighbours are usually easier to locate correctly as the potential for cooperation in localisation and error minimisation is greater, which is why the density of regular nodes also plays a significant role in overall accuracy. On the other hand, solutions which avoid the use of anchor nodes rely heavily on the nodes having the ability to estimate distances among themselves with significant accuracy, or the network topology being uniform enough that the variance in the relationship between actual distance and the hop count metric is low.

2.5 Summary and Problem Definition

Several conclusions may be drawn from the review of literature in the field of localisation in wireless networks. Infrastructure-based networks, be it cellular networks or WLANs, are completely dependent on their infrastructure, and as such, it seems a natural direction for the localisation algorithms aimed to operate in these networks to make use of this infrastructure. The benefit of having network infrastructure in place is that it offers, with more or less adjustment to the basic network architecture, a powerful system for performing, storing and analysing a number of different measurements on the scale of all nodes in the network, regardless of the type of localisation approach taken. The network can also then compute or provide aid in computing the node location estimates.

Ad hoc network, on the other hand, given their infrastructureless nature, rely completely on direct communications between the nodes. This makes them inherently void of any sort

of support when it comes to determining node locations. If no localisation infrastructure is placed in the network, such as anchor nodes, the information nodes can gather themselves through direct communication needs to be used in a centralised way, as is the case with anchor-free algorithms, e.g. gathering the set of all distance estimates between neighbouring nodes pairs, and solving the system of equations for unknown node locations given these constraints.

It can be concluded that, for the above-mentioned reasons, the existing solutions for node localisation in ad hoc wireless networks make one, or both, of the following assumptions:

1. There exist some reference points (anchors) in the network whose location is known.

2. There exists a mechanism for nodes to (relatively) precisely measure the distance to neighbouring nodes.

Furthermore, even the nominally decentralised localisation solutions for ad hoc networks are never decentralised to the level of an individual node, i.e. they only consider dividing the network into regions and running the same centralised algorithm on these smaller parts of the network. This means that nodes are always relying on some global set of information, obtained by either all other nodes in the network or at least their neighbourhood.

Having the previous in mind, the problem which this thesis is proposing a solution for can be formulated as *determining locations of single hop neighbours in wireless ad hoc networks of moving nodes, without the use of external reference signals (like GPS) or systems (like anchor nodes).*

The next chapter deals with the theoretical background of the proposed localisation solution, its application in a localisation algorithm in a realistic scenario, and the detailed description of the proposed *Relative Neighbour Localisation* (RNL) algorithm and each of its elements.

Chapter 3

Decentralised Localisation Algorithm Design

Drawing on the conclusions made in the previous chapter about the lack of a distributed, anchor-free localisation algorithm for ad hoc networks of moving nodes, the *Relative Neighbour Localisation* (RNL) algorithm is proposed. The goal of this algorithm is to enable nodes to estimate the locations of their neighbouring nodes in a highly dynamic environment using only signal strength measurements and the neighbouring nodes' mobility parameters, without the use of any external localisation signals or systems. It is based on the relationship of the trends in signal strength and the direction and speed of movement of the neighbouring node on one side and the relative location of that node on the other. The RNL algorithm estimates the relative angle and the distance at which the neighbouring node is located independently and then produces the two-dimensional location estimate from them.

This chapter first presents an introduction about the assumed network model, the theoretical principle of localisation using the above-mentioned algorithm inputs (signal strength and mobility parameters) and how the relationship between these inputs and the relative neighbour location can be estimated under the influence of the uncertainties present in a

realistic case. It then presents the RNL algorithm, defining each of its parts and the process of estimate computing in detail.

3.1 Introduction

3.1.1 Network model

The network model which is adopted for the remainder of this thesis assumes the following:

- mobile nodes which are moving completely independently of each other,
- nodes are assumed to move in paths composed of straight line segments,
- no global information about the location of the nodes exists; only relative locations are considered,
- the network is fully distributed, there is no hierarchy or clustering,
- each node is aware of its speed and direction of movement (i.e. is equipped with an accelerometer and compass),
- each node is broadcasting its velocity vector (the speed and the direction of movement) to its one-hop neighbours.

The constraints imposed by the decentralised approach to localisation are reflected on the choice of parameters used in order to estimate node locations and imply the use of values obtainable by each node individually as algorithm inputs. The most obvious choice for a parameter is *signal strength* as it is something any node is capable of measuring on its own and, in theory, depends on the distance between the sender and the receiver. The analysis of the fluctuations in the signal strength caused by the radio wave propagation effects and the imprecision and inaccuracy of the measurements are outside the scope of this work and are something that is taken as given.

Two assumptions about node mobility are made. The first one is about each node having knowledge of its velocity vector and broadcasting it periodically. This means that nodes are aware of neighbouring nodes speed and direction of movement, and can use this *velocity vector* as the second algorithm input. The second assumption is that nodes' velocity vector can be considered constant for considerably long periods of time, and only changing at discrete time instances.

In the interest of simplifying the analysis of the localisation principle, it will be assumed that node A, which is estimating a neighbouring node's location is stationary, and that only the neighbouring node B is moving. Figure 3.1, where $\overrightarrow{s_A}$ and $\overrightarrow{s_B}$ are vectors describing the movement of nodes A and B over a unit time period respectively, (i.e. $\overrightarrow{d_A} = 1\,\text{s} \times \overrightarrow{v_A}$, where $\overrightarrow{v_A}$ is the velocity vector of node A), shows the transformations allowing this perspective shift from a scenario of both nodes moving to one where node A is stationary. This also means that the conclusions of this analysis can be applied back to the case when both nodes are moving, using the relation shown in (3.1).

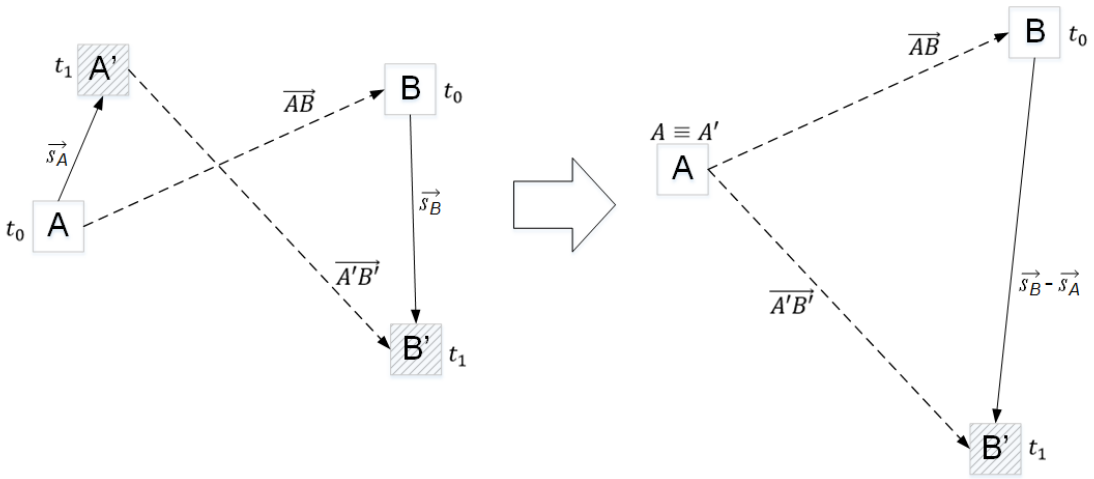


Fig. 3.1 Relation between velocity and position vectors

$$\begin{aligned} \overrightarrow{s_A} + \overrightarrow{A'B'} &= \overrightarrow{AB} + \overrightarrow{s_B} \\ \overrightarrow{A'B'} &= \overrightarrow{AB} + \underbrace{\overrightarrow{s_B} - \overrightarrow{s_A}} \end{aligned} \tag{3.1}$$

3.1.2 Ideal case localisation

This section presents the process of localisation of a neighbouring node in an idealised scenario. The following analysis derives expressions for the relative angle at which the neighbouring node is located (the *location angle*) and the distance at which it is located using measured signal strength and the neighbour's mobility parameters

Considering, for the moment, an ideal case of free space wireless radio communication, ignoring radio wave propagation effects, making signal strength purely a function of distance between the sending and receiving node. Using the Friis transmission formula [96], the ratio between the emitted (P_t) and received (P_r) signal strength can be expressed as:

$$\frac{P_r}{P_t} = D_t D_r \left(\frac{\lambda}{4\pi d} \right)^2, \tag{3.2}$$

where D_t and D_r are the directivities of the transmitting and receiving antennae respectively, λ is the wavelength and d is the distance between the sending and receiving nodes. Assuming that the transmission power is known, and that the antennae directivities are also known, the distance d between the two nodes can be expressed as:

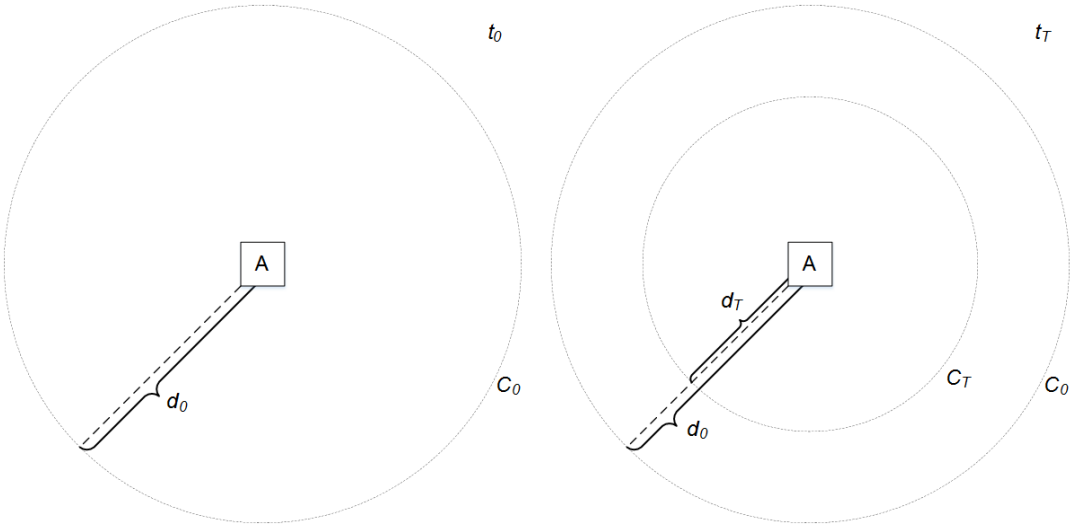
$$d = \frac{\lambda}{4\pi} \sqrt{\frac{P_t D_t D_r}{P_r}}. \tag{3.3}$$

Figure 3.2 represents the process of node A trying to determine the location of node B. Stationary node A receives the first broadcast of velocity vector $\overrightarrow{v_B}$ from moving node B at time $t = 0$ (Figure 3.2a) and measures the signal strength as ss_0, where ss_i is the level of

signal (ss) at time $t = i$ expressed in decibels-milliwatts, calculated as:

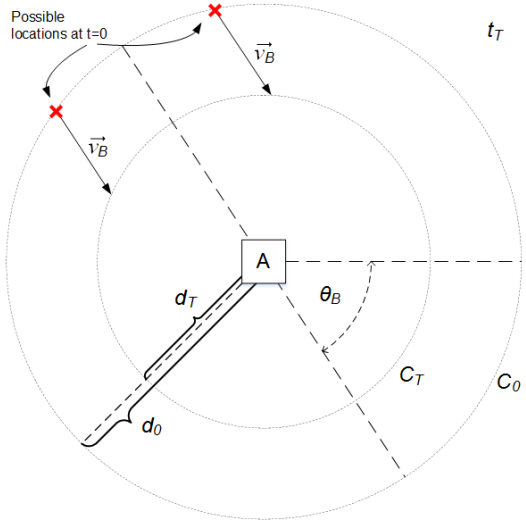
$$ss = 10 \log \frac{P_r}{1mW}. \tag{3.4}$$

At this point node A has an estimate of the distance d_0 to node B from ss_0, using (3.3). This places the possible locations of node B on a circle of radius d_0 with a centre in node A, denoted C_0.

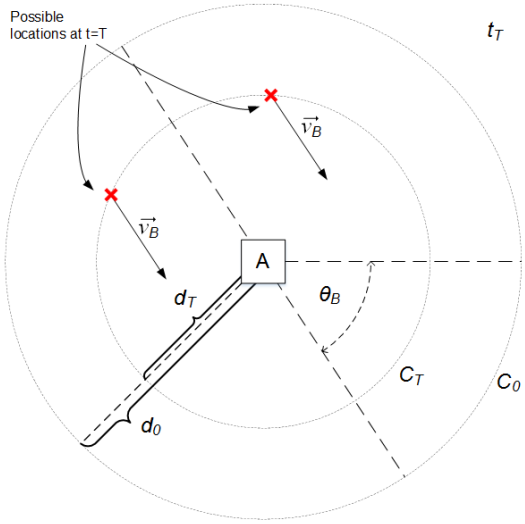


(a) First measurement

(b) Second measurement



(c) Possible movements of node B



(d) Possible locations of node B

Fig. 3.2 Localisation in the ideal case

At time $t = T$, where T is the interval of the velocity vector broadcasts, node A receives another broadcast from node B and measures the signal strength as ss_T, which translates to distance d_T (Figure 3.2b). Based on the new distance the location of node B must be somewhere on the circle of radius d_T with centre in node A, denoted C_T. Observing the possible locations at times $t = 0$ and $t = T$ and the velocity vector $\overrightarrow{v_B}$ narrows down the location of node B at $t = T$ to maximum of two points on C_T (Figure 3.2c and Figure 3.2d), as that is the number of points on C_0 from which vector $\overrightarrow{v_B}$ points to C_T.

This can be formalised analytically, with the following set of equations. A common reference (or zero angle) is assumed, and all other angles are measured with respect to it. The following notation is used: s is the distance covered by node B over a period of T seconds (i.e. $s = T \times |\overrightarrow{v_B}|$), θ_i is the direction of the velocity vector of node B at time $t = i$, and α_i is the angle of the line between node A and node B at time $t = i$. Figure 3.3 represents these values and their relation to distances d_0 and d_T.

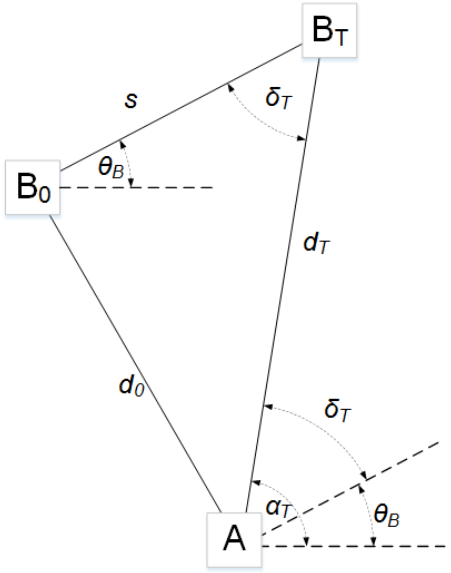


Fig. 3.3 Calculating the angle α

From the trigonometrical law of cosines for the triangle shown in Figure 3.3:

$$d_0{}^2 = d_T{}^2 + s^2 - 2 d_T s \cos \delta_T, \tag{3.5}$$

the value of angle δ_T can be derived as:

$$\delta_T = \arccos\left(\frac{{d_T}^2 + s^2 - {d_0}^2}{2{d_T}^2}\right). \tag{3.6}$$

This equation has two solutions, denoted $\delta_{T_{1,2}}$. Given this ambiguity, the value of angle α_T is also ambiguous and calculated as:

$$\alpha_{T_{1,2}} = \theta_T + \delta_{T_{1,2}} \tag{3.7}$$

which is shown in Figure 3.3 as well. The two values of α correspond to one of the points from Figure 3.2d each. Only one of them represents the *true* location of node B at time T, while the other one represents what will be referred to as the *symmetrical* location of node B at time T. The symmetry between these two point is exhibited with respect to the line defined by the position of node A and the angle θ_T. The relation between the ambiguous relative angle values (shown in Figure 3.4) is the following:

$$\begin{aligned} \alpha_{T_1} &= \theta_T - (\alpha_{T_2} - \theta_T) \\ \alpha_{T_1} &= 2\theta_T - \alpha_{T_2}. \end{aligned} \tag{3.8}$$

At this point, the ambiguous relative location of node B, i.e. its location in the Cartesian coordinate system with node A in origin, is computed as:

$$\begin{aligned} x_{T_{1,2}} &= d_T \cos\alpha_{T_{1,2}} \\ y_{T_{1,2}} &= d_T \sin\alpha_{T_{1,2}}. \end{aligned} \tag{3.9}$$

In order to resolve this ambiguity, and reduce it into the true location of the node, more information is needed. This information is obtained from a change in direction of movement of node B. Once a broadcast with a different velocity vector, with direction θ_{2T} is received

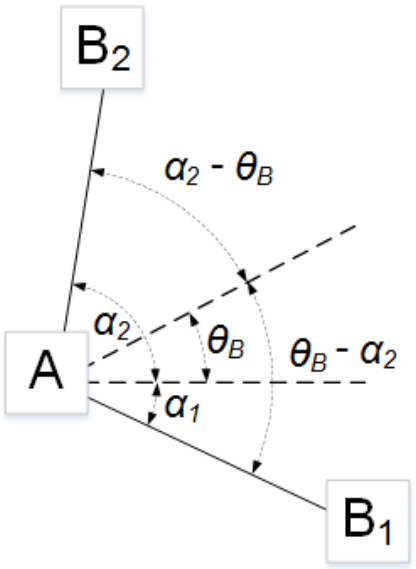


Fig. 3.4 Ambiguous angle symmetry

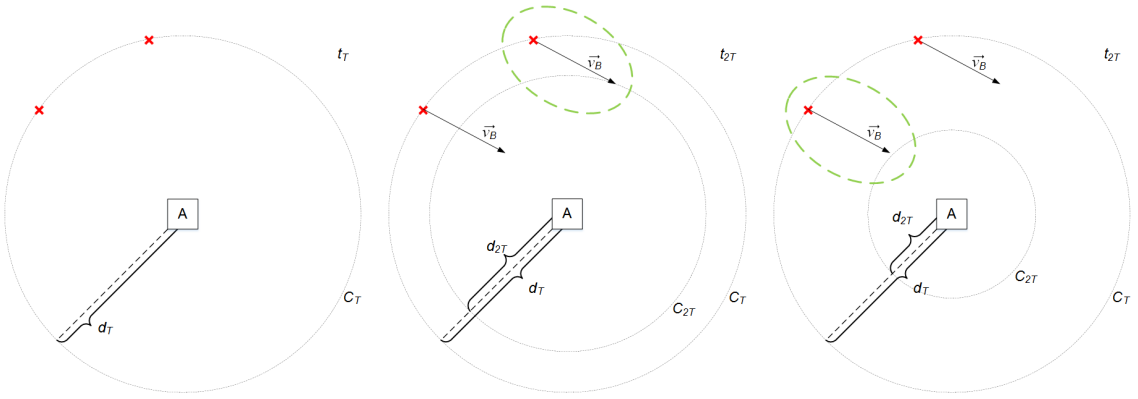
at time $2T$ from node B, node A can, based on possible locations from the previous step, determine the two possible distances to node B at time $2T$ as:

$$\begin{aligned} \hat{d}_{2T_1} &= \sqrt{{d_T}^2 + s^2 + \cos(\alpha_{T_1} - \theta_{2T})} \\ \hat{d}_{2T_2} &= \sqrt{{d_T}^2 + s^2 + \cos(\alpha_{T_2} - \theta_{2T})}, \end{aligned} \tag{3.10}$$

and compare the distance at $2T$ obtained from ss_{2T}. If $d_{2T} = \hat{d}_{2T_1}$, the angle α_{T_1} was the real angle and if $d_{2T} = \hat{d}_{2T_2}$, the angle α_{T_2} was the real one, and the ambiguity is resolved, allowing node A to continue tracking the unique location of node B. This procedure is shown in 3.5, where 3.5a shows two possible locations at $t = T$, and 3.5b and 3.5c show the situation at $t = 2T$, after a turn has been detected and the new distance has been estimated, with possible locations one and two having been recognised as the real location, respectively.

3.1.3 Real case limitations

The analysis presented above is based on the assumptions of ideal free space propagation, no interference and ideal measurements of the parameters involved available. In order to show



(a) Ambiguous location at $t = T$

(b) Possible location one selected as the real one

(c) Possible location two selected as the real one

Fig. 3.5 Resolving the ambiguous location after a turn

the applicability of the described relation between the change in signal strength, mobility parameters and relative locations of two nodes, it should be investigated in a generalised, realistic case.

By removing the ideal free space signal propagation assumption the presented localisation principle is faced with its first obstacle. Real world signal propagation means that the signal strength will be a function of factors other than just the distance between the sending and the receiving node. Different obstacles and environment configuration can influence the path of radio waves, causing them to reach the receiver via multiple paths. *Multipath propagation* and *signal shadowing* are responsible for fluctuations in the strength of received signal, known as *fading*. The effects of fading on measures signal strength are shown in Figure 3.6, which shows the received signal strength fluctuations over time.

The second assumption made in the ideal scenario is having absolutely precise measurements of signal strength and speed and direction of movement. In a realistic scenario the devices measuring these parameters are either not sensitive enough, not precise enough, or both.

The imperfections of the real world scenario complicate the relationship between the distance and received signal strength, and the expression from (3.3) can no longer be assumed

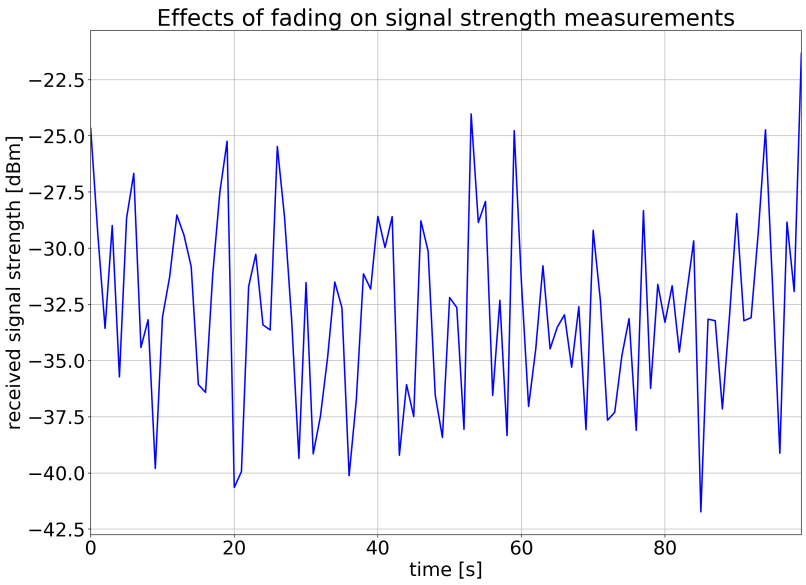


Fig. 3.6 Effects of fading on signal strength measurements

to hold. While a number of experimental models have been proposed based on averaging the signal strength measurements from a transmitter at a known distance over time, thus stabilising this relationship and producing relatively accurate distance estimates [97–99], a certain error is unavoidable. This makes it impossible to apply the idealised approach described in the previous section, especially having in mind the mobility of both the receiver and transmitter, as even small errors in two consecutive distance estimates shown in Figure 3.3 could result in large errors of the calculated location angle.

Having these hindering factors in mind, a conclusion can be drawn that localisation in a realistic case based on the dependency between the signal strength, the node mobility and the relative locations described in the previous subsection can be expected to produce less precise results, or in other words, there would be more *uncertainty* compared to the ideal case. The effects of fading along with the lack of absolutely precise measurements all add to the *unreliability* of location estimation.

Given the network model, the presented theoretical basis on which the proposed localisation algorithm should work and the limitations on the realistic implementations of it, the research problem for this PhD work can be more precisely defined as follows: *determining*

single-hop neighbour relative location relying only on its mobility parameters (velocity vector) and the performed signal strength measurements.

3.2 Algorithm Overview

Given the uncertainties present in a realistic scenario, it would be very difficult, if not impossible, to derive an analytical form of the localisation algorithm, such as the one shown in section 3.1.2. It is, however, assumed that even with the effects of fading and imprecise measurements, trends in the signal strength measurement can still be identified by smoothing them over time using moving averages. This means that the relation between the signal strength and the velocity vector can still be utilised for location estimation in a non-ideal scenario.

An empirical approach is taken to characterise this complex relation, by using a regression algorithm called *Random Forest Regression* (RFR) as the basis of the angle and distance estimation procedures. This means that the algorithm needs to go through an offline, learning phase, in which it is fed measurements based on which it learns about the relations between the measurements available as inputs and the desired outputs. The reasons for choosing this learning algorithm over others will be presented in section 4.3.

The training of the estimators can be performed using real measurements with known true locations of target nodes. The *input – output* pairs obtained in such way, are provided to the untrained estimators, which analyse the dataset and form mappings from the input values to the desired outputs. Another method for obtaining a training dataset can be by simulation methods, carefully choosing the network and environment parameters in such a way that they correspond to the scenario in which the application of the algorithm is intended. Both the experimental and simulation datasets have been obtained, details of which are presented in sections 4.2.2 and 4.2.4.

3.2.1 Algorithm structure

The *Relative Neighbouring Localisation* (RNL) algorithm is composed of three parts, or modules, which can be put together incrementally, to produce the final location estimate. The first part of the algorithm is the *location angle estimation*, which is composed of producing what is called a *raw angle estimate* and resolving the *symmetrical angle issue*. The second part of the RNL algorithm is the *distance estimation*. Given the outputs of these two parts, a two-dimensional *location estimate* is computed. The third part of the algorithm is *estimate sharing*. This part, although violating the strict definition of the distributed nature of the proposed algorithm, is introduced as a method of increasing localisation accuracy, and is confined to one step neighbours exchanging their location estimates of nodes in their range. Given the characteristics of the location estimates created by the first two steps of the RNL and the estimates received through the third step, a final estimate is produced. Figure 3.7 shows the block diagram representation of the full RNL algorithm with its inputs and outputs.

Throughout the rest of this chapter, all the algorithm inputs and outputs will be regarded in a discrete time domain, with time instances denoted as indexes in square brackets. The node whose location is being estimated will be referred to as the *target node*, the node performing the RNL algorithm will be referred to as just *root*, and the nodes sharing their estimates with root will be referred to as *neighbours*. Superscripts will be used to denote the source of a particular estimate or measurement (if it is not the root node), i.e. which node has produced it, and subscripts will be used to denote the node estimates are relating to (if it is not the target node). The following notation will be used for certain values and measurements:

ss Measured received target node signal strength,

ss_{avg} Average measured received target node signal strength,

n_{avg} Signal strength averaging window size,

θ Direction of movement of the target node, relative to root,

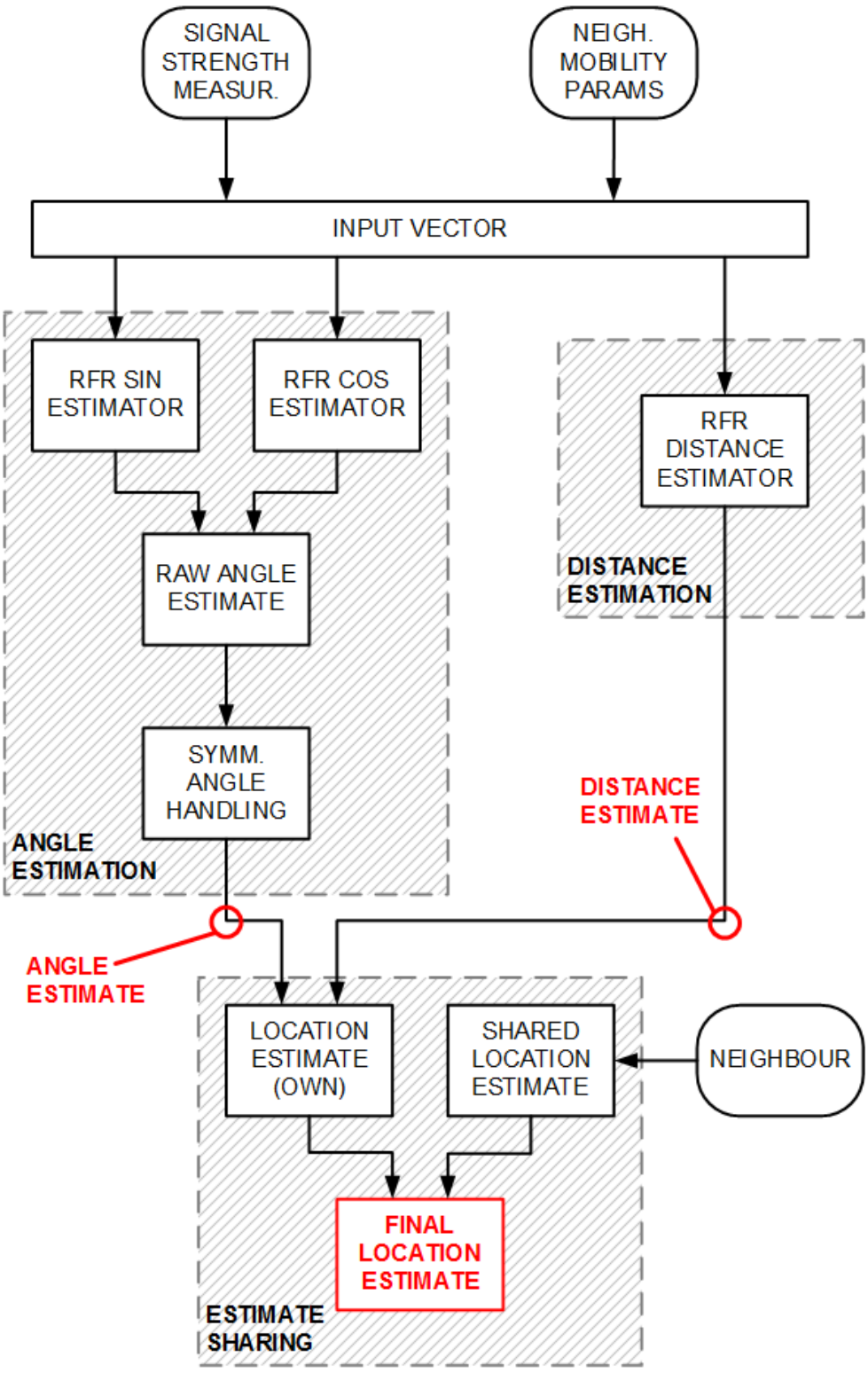


Fig. 3.7 RNL block diagram

s Speed of movement of the target node, relative to root,

α True location angle of the target node,

γ_R, γ_S Raw location angle estimate produced by the RFR, and its symmetrical angle,

n_{prev} Number of previous averaged signal strength values used as RFR inputs,

$\hat{\alpha}_1, \hat{\alpha}_2$ Ambiguous location angle estimates,

$\hat{\alpha}$ Unique location angle estimate (after ambiguity is resolved),

d True distance of the target node,

$\hat{d}$ Estimated distance of the target node,

x, y True target node coordinates, relative to root,

x^i, y^i True target node coordinates, relative to neighbour i,

$\hat{x}, \hat{y}$ Estimated target node coordinates relative to root,

$\hat{x}^i_{sh}, \hat{y}^i_{sh}$ Estimated target node coordinates received from node i, relative to root,

$\hat{x}_{sh}, \hat{y}_{sh}$ Estimated target node coordinates from all shared estimates.

The relative direction and speed of movement (θ, s) are calculated according to Figure 3.1 and equation (3.1), from the relative velocity vector of the target and root nodes $(\overrightarrow{v}_t, \overrightarrow{v}_r)$, as follows:

$$\begin{gathered} \overrightarrow{v} = \overrightarrow{v_t} - \overrightarrow{v_r} \\ s = \sqrt{(s_t \cos\theta_t - s_r \cos\theta_r)^2 + (s_t \sin\theta_t - s_r \sin\theta_r)^2} \\ \theta = \arctan\left(\frac{s_t \sin\theta_t - s_r \sin\theta_r}{s_t \cos\theta_t - s_r \cos\theta_r}\right). \end{gathered} \tag{3.11}$$

3.3 Angle Estimation

The estimation of the location angle of the target node is performed in two steps. The first step assumes producing what is called a *raw estimate*, using the RFR algorithm. The second step refers to the resolving (if possible) the location angle estimate ambiguity.

3.3.1 Raw RFR estimates

The RFR estimator works by producing an output value for a given set of input values, based on the relationships between these parameters and their values it has formed during the training stage. It takes the vector of n_{prev} averaged signal strength values as the first part of the input, while the second part are the target node mobility parameters:

$$input[t] = [ss_{avg}[t - n_{prev} + 1], \ldots, ss_{avg}[t-1], ss_{avg}[t], \theta[t], \sin\theta[t], \cos\theta[t], s[t]]. \quad (3.12)$$

The direction angle is present as both the exact value, as well as its sine and cosine values. This is done because the estimator is unaware of the periodic nature of the angular values, i.e. the fact that angles π and $-\pi$ are the same angle. To overcome this, trigonometric functions are used, as their values allow for these two angles to be perceived as equal.

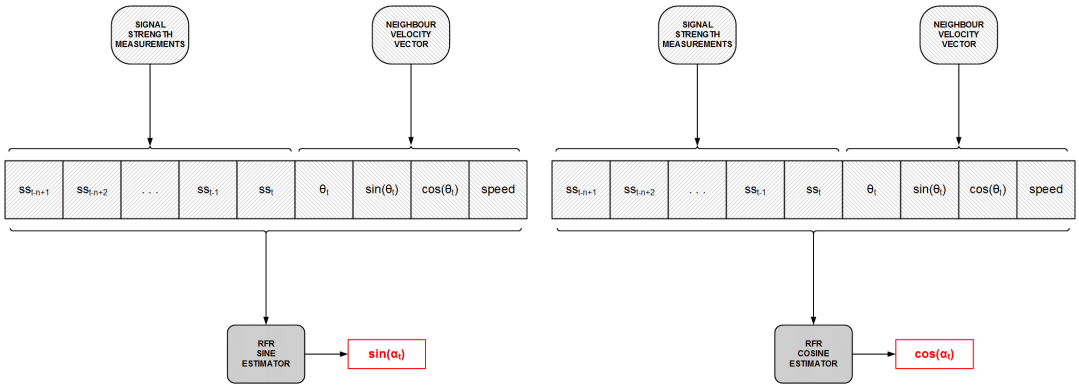
The vector $input[t]$ represents one input data point. During the estimator training phase, the estimator is presented with a dataset composed of *input – output* pairs of data points. The output presents the true value of the location angle, in the following form:

$$output[t] = [\sin\alpha[t], \cos\alpha[t]]. \quad (3.13)$$

Following the same reasoning concerning the periodicity of the angular values as above, the true location angle is represented by its sine and cosine values. Because of this, there are two estimators being used, one for the sine value est_{sin}, and the other one for the cosine value est_{cos}. The mappings that they are trained to estimate can be defined as:

$$\begin{aligned} est_{sin} &: input \rightarrow \sin\alpha \\ est_{cos} &: input \rightarrow \cos\alpha, \end{aligned} \quad (3.14)$$

and the two estimators are shown in Figure 3.8.



(a) Sine estimator (b) Cosine estimator

Fig. 3.8 The sine and cosine RFR estimators

During the online phase, after the estimators have been trained, the raw angle estimate $\hat{\alpha}$ is produced as:

$$\begin{aligned} \sin \gamma_R &= est_{sin}(input) \\ \cos \gamma_R &= est_{cos}(input) \\ \gamma_R &= \arctan\left(\frac{\sin \gamma_R}{\cos \gamma_R}\right). \end{aligned} \tag{3.15}$$

3.3.2 Symmetrical angle problem

As described in section 3.1.2, the presence of ambiguity of location angle estimates is inevitable even in the ideal case. This means that, given a set of input parameter values, there exist two location angles which could theoretically correspond to it. The estimators defined above have no ability to distinguish between these ambiguous values, as they are trained using the true output (location angle) values. In the process of training, the estimators develop a set of rules which map the input parameter values into the produced output estimation. These rules are an interpretation of the relationship between signal strength and mobility parameters on one hand and the relative location angle on the other. Even for perfectly trained estimators, the ambiguity of the location angle still exists, as the two possible location angles share the same relation to the input parameters. The estimator producing a symmetrical instead of

the true location angle would still not be wrong, as it would have applied these relations correctly. Figure 3.9 shows an example of RNL raw and symmetrical angle estimates.

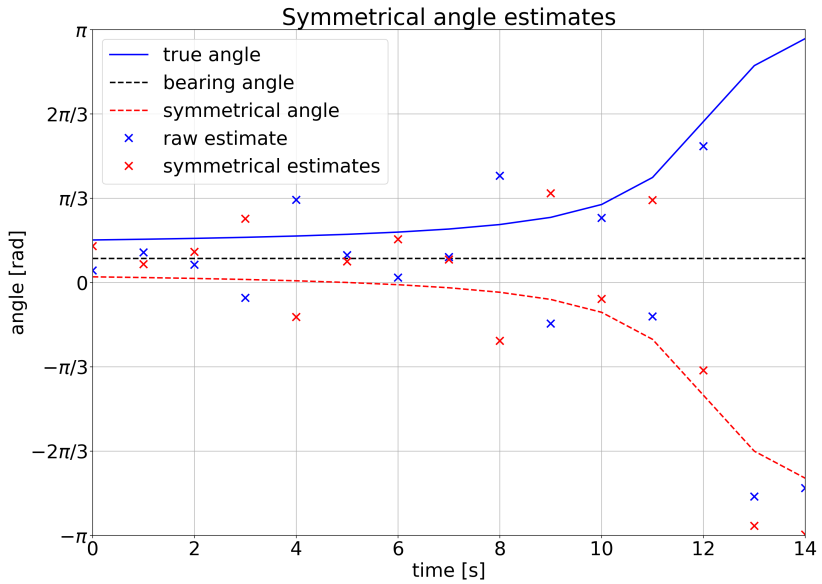


Fig. 3.9 RNL symmetrical angle estimates

RNL algorithm deals with the location angle ambiguity by maintaining two variables $\hat{\alpha}_1$ and $\hat{\alpha}_2$ for each of the two possible values of the location angle. Once the localisation is initiated, and the first RFR raw estimate γ_R of the target node's location angle is produced, this value is assigned to $\hat{\alpha}_1$. Using equation (3.8), the symmetrical value γ_S is calculated, and assigned to $\hat{\alpha}_2$. For every consecutive pair of the raw and symmetrical RFR estimates $(\gamma_R[t], \gamma_S[t])$, the following four differences are calculated:

$$\begin{aligned} \delta_{1R} &= |\hat{\alpha}_1[t-1] - \gamma_R[t]| \\ \delta_{1S} &= |\hat{\alpha}_1[t-1] - \gamma_S[t]| \\ \delta_{2R} &= |\hat{\alpha}_2[t-1] - \gamma_R[t]| \\ \delta_{2S} &= |\hat{\alpha}_2[t-1] - \gamma_S[t]|. \end{aligned} \tag{3.16}$$

These differences represent the distance between the new estimates and the previous ones. As the nodes are assumed to be moving in straight lines, the location angle cannot

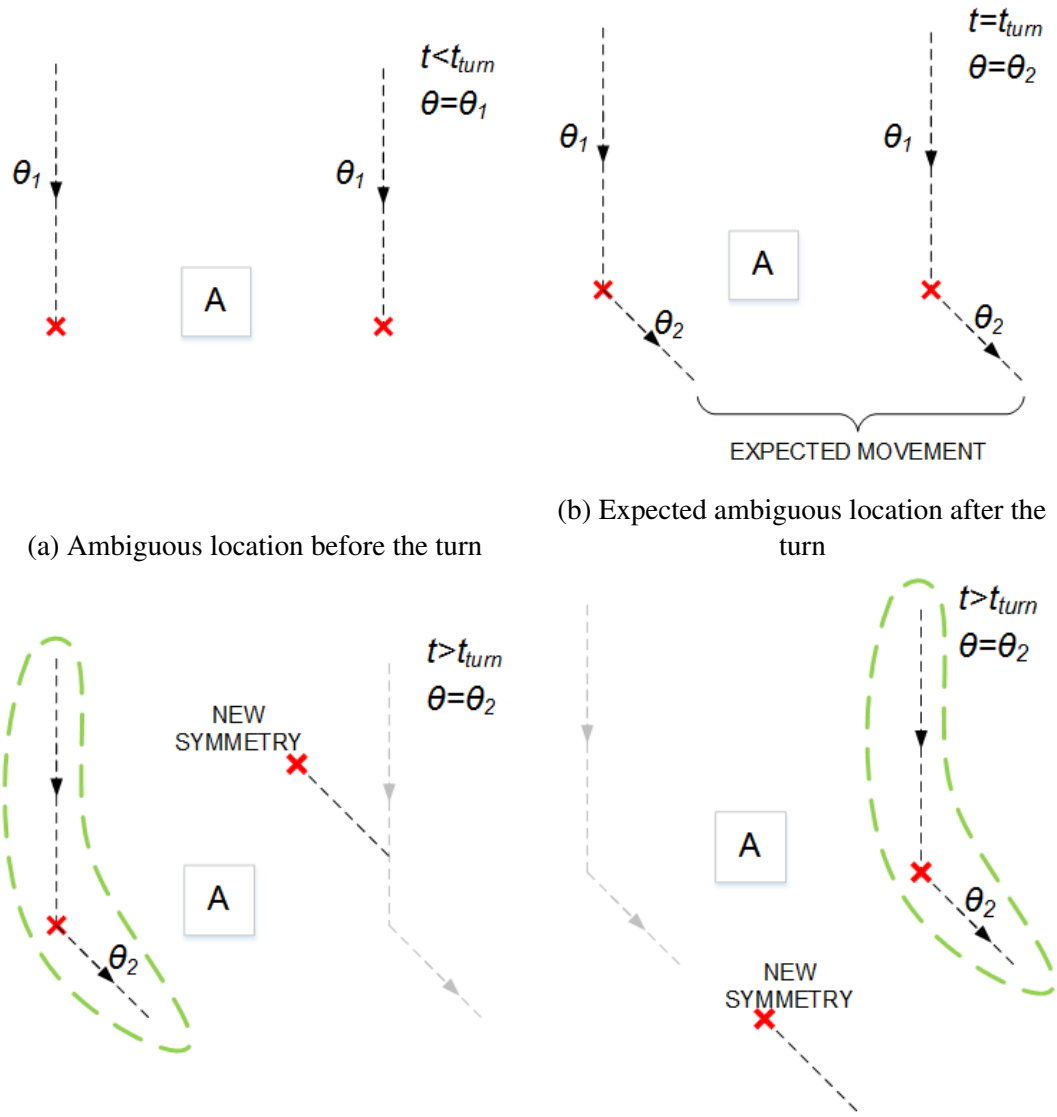
change significantly between two estimates, and it is assumed that the difference between the symmetrical angle and the previous true angle values would be greater than the difference between two consecutive true angle values. This comparison is referred to as *type 1*. Consequently, if the smallest of the four differences is δ_{1R} or δ_{2S}, that means that the raw RFR estimate $\gamma_R[t]$ is the one corresponding to the $\hat{\alpha}_1[t-1]$ angle estimate, and the symmetrical RFR estimate $\gamma_S[t]$ corresponds to the $\hat{\alpha}_2[t-1]$ estimate. The new values of the two angle estimates are therefore assigned as:

$$\begin{aligned} \hat{\alpha}_1[t] &= \gamma_R[t] \\ \hat{\alpha}_2[t] &= \gamma_S[t]. \end{aligned} \tag{3.17}$$

If the opposite is true, i.e. if the smallest of the four differences is calculated to be either δ_{2R} or δ_{1S}, the new angle estimates are given the following values:

$$\begin{aligned} \hat{\alpha}_1[t] &= \gamma_S[t] \\ \hat{\alpha}_2[t] &= \gamma_R[t]. \end{aligned} \tag{3.18}$$

According to the analysis presented in section 3.1.2, the location angle ambiguity can be resolved with additional information obtained after the target node makes a change in the direction of movement. This procedure, from the perspective of the symmetrical angle handling part of the angle estimation element of the RNL algorithm, is shown in Figure 3.10, where Figure 3.10a shows the two locations corresponding to the angle estimates before the turn, $\hat{\alpha}_1$ and $\hat{\alpha}_2$. Once a change of direction happens, the RFR estimator produces a raw estimate based on the new input values. At the same time, given the new mobility parameters, the target node is expected to be located in one of the two locations shown in Figure 3.10b. This time, the axis of symmetry will be different, resulting in a mismatch between the new γ_R and γ_S values on one side and the previous $\hat{\alpha}_1$ and $\hat{\alpha}_2$ values on the other.



(a) Ambiguous location before the turn

(b) Expected ambiguous location after the turn

(c) New estimate leads to discarding the right estimate, left one adopted as true

(d) New estimate leads to discarding the left estimate, right one adopted as true

Fig. 3.10 Symmetrical angle ambiguity

Using the calculations shown in equation (3.16), the smallest difference between the previous and the new estimates can be found. This time, the smallest difference determines which of the previous location angle estimates was the true one, and which of the newly produced RFR estimates corresponds to it. This comparison is referred to as *type 2*. At this point, the $\hat{\alpha}_1$ and $\hat{\alpha}_2$ are discarded, and a unique angle estimate $\hat{\alpha}$ is introduced, according to

the following rule:

$$\begin{aligned}
\Delta &= \{\delta_{1R}, \delta_{1S}, \delta_{2R}, \delta_{2S}\} \\
\text{if} \min(\Delta) = \delta_{1R} &\implies \hat{\alpha}[t-1] = \hat{\alpha}_1[t-1], \hat{\alpha}[t] = \gamma_R[t] \\
\text{if} \min(\Delta) = \delta_{1S} &\implies \hat{\alpha}[t-1] = \hat{\alpha}_1[t-1], \hat{\alpha}[t] = \gamma_S[t] \\
\text{if} \min(\Delta) = \delta_{2R} &\implies \hat{\alpha}[t-1] = \hat{\alpha}_2[t-1], \hat{\alpha}[t] = \gamma_R[t] \\
\text{if} \min(\Delta) = \delta_{2S} &\implies \hat{\alpha}[t-1] = \hat{\alpha}_2[t-1], \hat{\alpha}[t] = \gamma_S[t].
\end{aligned} \tag{3.19}$$

Figures 3.10c and 3.10d show the two cases in which one of the two possible locations is discarded based on the rule defined above.

From this point on, the choice between the raw and symmetrical RFR estimates is made by comparing them to the last value of the unique angle estimate $\hat{\alpha}$ as:

$$\hat{\alpha}[t] = \begin{cases} \gamma_R[t], & \text{if } |\gamma_R[t] - \hat{\alpha}[t-1]| \leq |\gamma_S[t] - \hat{\alpha}[t-1]| \\ \gamma_S[t], & \text{if } |\gamma_R[t] - \hat{\alpha}[t-1]| > |\gamma_S[t] - \hat{\alpha}[t-1]| \end{cases} \tag{3.20}$$

This comparison is referred to as *type 3*. Block diagram of the whole symmetrical angle handling procedure is shown in Figure 3.11.

3.4 Distance Estimation

The distance to the target node in the RNL is estimated using the same mechanism used in the angle estimation. During the training phase, the training dataset, composed of *input – output* pairs, is presented to the distance estimator. The input vector used is the same one used in the angle estimation, while the output is the true distance between the root and the target nodes. Although only one of the input parameters, $ss[t]$, can be directly linked to the distance between the two nodes, the use of additional information has been shown to lead to

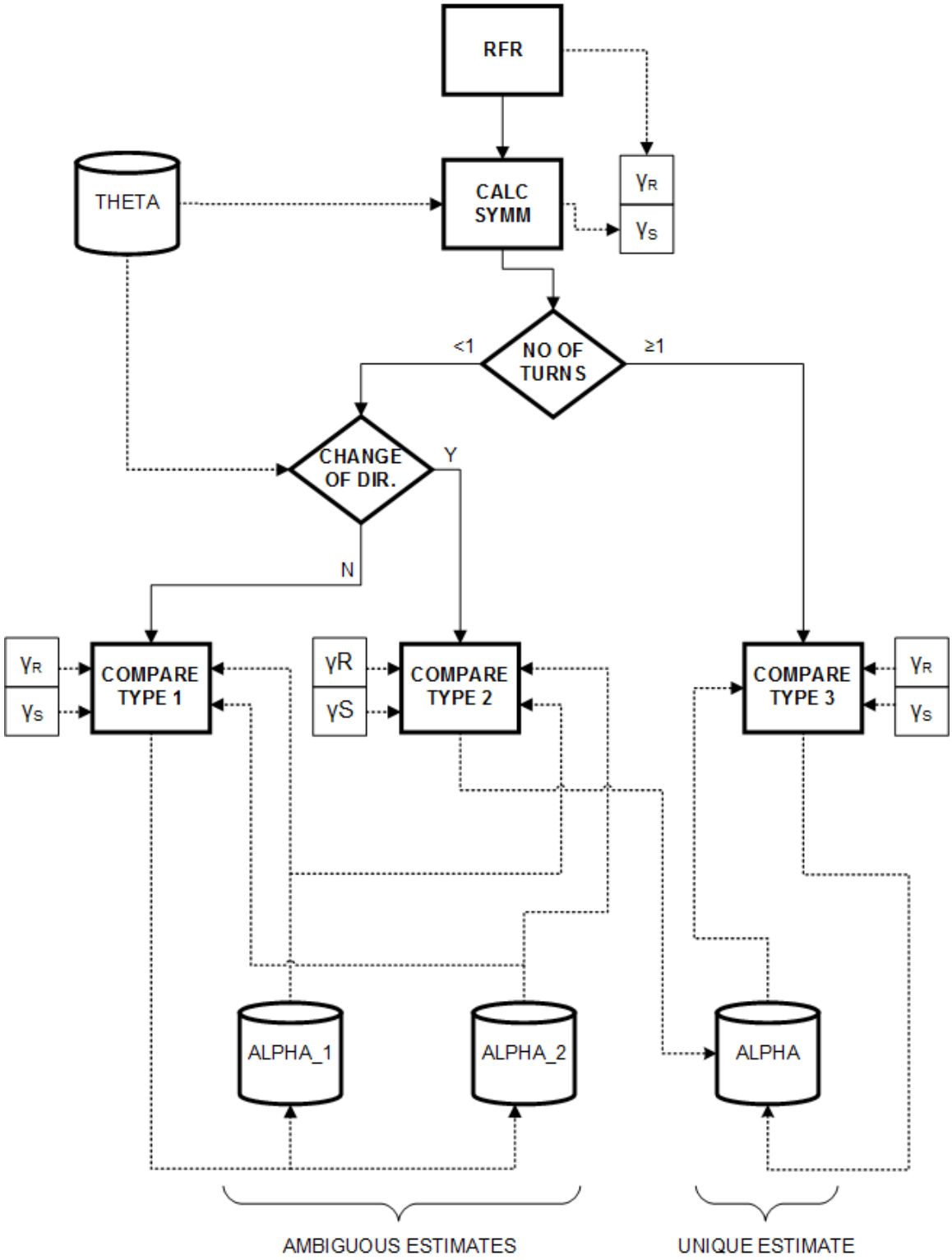


Fig. 3.11 Symmetrical angle handling block diagram

better accuracy in distance estimation. This is further discussed in the next chapter, along with other evaluation results of the RNL performance. The distance estimator, est_{dist}, is trained to perform the following mapping:

$$est_{dist} : input \rightarrow d, \tag{3.21}$$

and is shown in Figure 3.12. During the online phase, after the estimator has been trained, the distance estimate $\hat{d}$ is produced as:

$$\hat{d} = est_{dist}\,(input)\,. \tag{3.22}$$

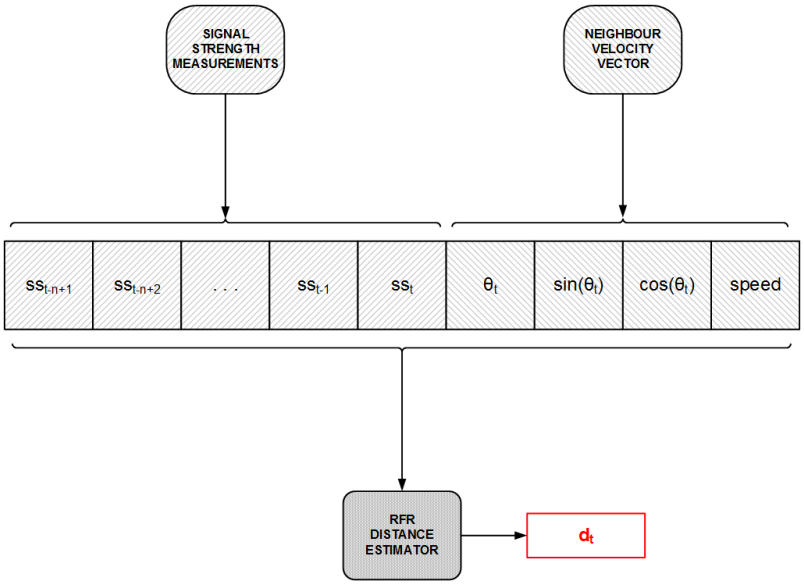


Fig. 3.12 Distance estimator

3.5 Estimate Sharing

Using the angle and distance estimators defined so far, the root node is able to compute the relative $\hat{x}, \hat{y}$ coordinates of the target node. Given the distributed nature of the RNL algorithm and the uncertainty of the complex relationship between the available measurements used as algorithm inputs and the location of the target node, the proposed localisation method cannot be expected to produce high accuracy.

As an attempt to increase the accuracy of the location estimates, an estimate sharing procedure is defined. The estimate sharing among single-hop neighbours is designed to include group knowledge into the localisation process, and to detect and possibly avoid some large errors, by analysing and comparing target node location estimates of multiple neighbours to the ones locally produced using the angle and distance estimators.

In general, the estimate sharing procedure assumes each node computing its own local estimates of the relative locations of all of its neighbouring nodes and sharing them, using periodic broadcasts, with that same group of single-hop neighbours. In the interest of simplicity, this process will be described assuming there is only one target node whose location estimates are being shared by nodes whose range it is within.

3.5.1 Computing shared estimates

Each node, before engaging in the estimate sharing procedure, calculates the target node location estimates relative to itself. Given the angle and distance estimates, the relative location of the target node is calculated as:

$$\begin{aligned} \hat{x}[t] &= \hat{d}[t]\cos\left(\hat{\alpha}[t]\right) \\ \hat{y}[t] &= \hat{d}[t]\sin\left(\hat{\alpha}[t]\right). \end{aligned} \tag{3.23}$$

In case the location ambiguity has not yet been resolved, two target node location estimates are kept:

$$\begin{aligned} \hat{x}_{1,2}[t] &= \hat{d}[t]\cos\left(\hat{\alpha}_{1,2}[t]\right) \\ \hat{y}_{1,2}[t] &= \hat{d}[t]\sin\left(\hat{\alpha}_{1,2}[t]\right). \end{aligned} \tag{3.24}$$

Once own relative location estimates have been computed, each node broadcasts its unique or ambiguous location estimate of the target node. As the root node receives these broadcasts, it needs to take into consideration its own relative location estimates of the node which is the source of the broadcast, which may also be unique or ambiguous. Assuming unique location estimates in both of the above cases, and node a sharing its target node location estimate with the root, the root node calculates the shared estimate in its own coordinate system as:

$$\begin{aligned} \hat{x}^a_{sh} &= \hat{x}_a + \hat{x}^a \\ \hat{y}^a_{sh} &= \hat{y}_a + \hat{y}^a, \end{aligned} \tag{3.25}$$

where $(\hat{x}_a, \hat{y}_a)$ is the root's estimate of the location of node a, and $(\hat{x}^a, \hat{y}^a)$ is the target node location estimate produced by node a. In cases where one or both of these location estimates are ambiguous, this process can result in two or four different coordinate pairs respectively.

A set of all target node location estimates, $\mathbb{E}$, consisting of the root node's own and the estimates received and calculated through the process of sharing, is created as:

$$\mathbb{E} = \{(\hat{x}_t, \hat{y}_t)\} \cup \left\{ \left(\hat{x}^j_{sh}, \hat{y}^j_{sh} \right) \forall j \in \mathbb{S}, j \neq t \right\}, \tag{3.26}$$

where $\mathbb{S}$ denotes the set of root node's neighbours and t is the target node.

3.5.2 Clustering

Given the inherent uncertainty with which the RNL operates, and which is in the basis of all the estimates produced, it may be assumed that not all of the estimates in the set $\mathbb{E}$ are close to the true location of the target node. This is especially so in the cases where location ambiguity is present, and even more so if there are four different estimates being associated with one source, as described in the previous subsection. If the nodes are producing relatively accurate estimates, they can be expected to be close to each other, while the estimates with low accuracy can be expected to be dispersed around different areas, as each node which is the source of an estimate observes different values of the target node's relative mobility parameters. This rationale can be used to draw a conclusion about which of the estimates may be the accurate ones, and which ones the erroneous ones, by analysing the shared estimates.

The analysis of the estimate set $\mathbb{E}$ is performed using a clustering procedure. More precisely, *hierarchical agglomerative clustering* is used. This process, described in [100], assumes the following steps:

1. Create single-element subsets (clusters) of the given set,
2. Unite two closest clusters,
3. Add the unity subset to the list of clusters; remove the two which it was created from,
4. Repeat steps 2 and 3 until there is only one cluster.

The closeness of the clusters is examined using Ward's minimum variance criterion, which ensures that the clusters selected for merging are the ones which will result in a new cluster with the smallest variance within its member elements.

As the goal of clustering in RNL estimate sharing is to determine if there is a close group of estimates, indicating that they are all high accuracy estimates, there is no need to run the above clustering procedure until the end. What is more interesting given the goal of this

procedure, is to find the smallest variance cluster of a particular size. This minimum desired cluster size, denoted as c, is relative to the cardinality of the set of all estimates $\mathbb{E}$, i.e. for a larger estimate set, a greater number of similar estimates is required for them to be assumed high accuracy, while for a smaller estimate set this number needs to be reduced in order for a low variance cluster with a given size to be possible to exist.

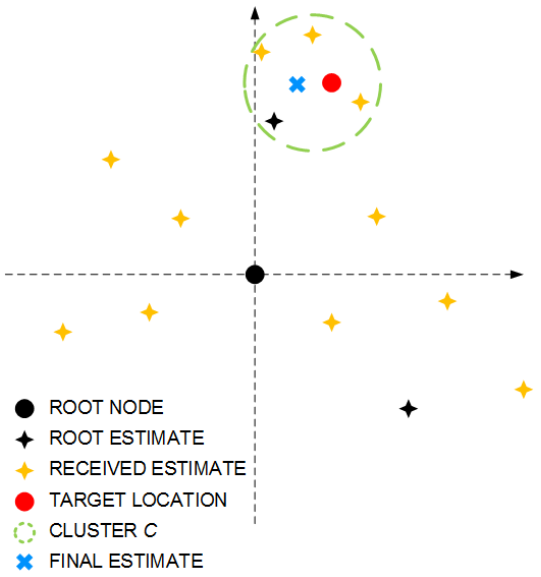


Fig. 3.13 Example of shared estimates and clusters

The first cluster of size c which is produced by the clustering procedure described above, is guaranteed to be the one with the smallest variance, and is referred to as the chosen cluster, and denoted as C. This cluster contains what are assumed to be the high accuracy target node location estimates, and the final target node location estimate can then be computed as the centroid of all estimates in C:

$$\begin{aligned} \hat{x}_{sh} &= \frac{\sum_j \hat{x}^j_{sh}}{|C|}, j \in C \\ \hat{y}_{sh} &= \frac{\sum_j \hat{y}^j_{sh}}{|C|}, j \in C. \end{aligned} \tag{3.27}$$

The clustering procedure is illustrated in Figure 3.13.

3.6 Special Case – Estimate Propagation

Both the angle and the distance estimation components of the RNL algorithm rely on the RFR estimators to produce their outputs. As seen above, the estimation process is defined to use n_{prev} signal strength measurements. This means that the RNL is unable to produce estimates for the first n_{prev} seconds after a new neighbour has appeared in range. In order to avoid not having a location estimate during these periods, shared estimates are used if present.

A similar issue arises when the relative mobility parameters of the target node change, i.e. either the root or the target node change their direction or speed of movement significantly. Once a change in relative mobility happens, the n_{prev} signal strength measurements no longer correspond to the current relative direction of movement of the target node, and should not be used as estimator inputs. In this case, if previous estimates produced by the root node exist, *estimate propagation* is performed. The last $\hat{x}, \hat{y}$ estimates are transformed using the relative mobility parameters according to the following expressions:

$$\begin{aligned} &\text{if } t - t_{last\_turn} < n_{prev} : \\ &\hat{x}[t] = \hat{x}[t-1] + s\cos\theta \\ &\hat{y}[t] = \hat{y}[t-1] + s\sin\theta . \end{aligned} \tag{3.28}$$

This way, periods of potential RNL estimate unavailability are avoided.

3.7 Location Tracking

The RNL algorithm produces estimates at each step independently from the previously generated estimates. This may cause consecutive estimates to be placed in a manner which does not correspond to the mobility parameters of the target node (i.e. consecutive estimates

placed at a distance or in a direction which is not possible given the relative speed and the relative bearing angle of the target node). This can have large location estimate errors as a result. In order to reduce the impact of this issue, location tracking can be applied.

The information about the target node mobility can be used to correct the location estimates produced by the RNL. This is done by fitting consecutive RNL location estimates into an array of estimates which correspond to the laws of movement based on the mobility information. Each new estimate produced by the RNL during a sequence of constant values of relative (between the root and the target node) mobility parameters is compared to all the previous estimates in that sequence, using the following analysis:

$$\begin{aligned} \hat{x}_{exp_k}[t] &= \hat{x}[t-k] + s\cos\theta \\ \hat{y}_{exp_k}[t] &= \hat{y}[t-k] + s\sin\theta, \end{aligned} \tag{3.29}$$

where $\hat{x}[t-k], \hat{y}[t-k]$ is the estimate produced k steps ago, and $\hat{x}_{exp_k}[t], \hat{y}_{exp_k}[t]$ is what the latest estimate is expected to be, based only on the relative mobility parameters and the estimate produced k steps ago.

For a sequence of l estimates, the set of following differences is calculated:

$$\varepsilon_{i,j} = |\hat{\mathbb{L}}_{exp_j}[i] - \hat{\mathbb{L}}[i-j]|, \;\; i,j \in \{0,\ldots,l\}, \;\; i < j, \tag{3.30}$$

where $\hat{\mathbb{L}}_{exp_j} = (\hat{x}_{exp_j}, \hat{y}_{exp_j})and\hat{\mathbb{L}} = (\hat{x}, \hat{y})$. The pair of estimates i, j for which the minimum value of $\varepsilon_{i,j}$ is obtained, is chosen as an *anchor* pair. These anchor points are the two estimates which are placed in such way that they best fit the relative mobility parameters and laws of motion, making them most likely to be the most accurate ones. A line in the direction of movement is drawn so that it is the best fit through the anchor points, and all

other estimates are then realigned along this line. This results in a set of estimates created over a single sequence corresponding to the relative mobility parameters of the target node.

3.8 Summary

The idea behind the localisation approach of the Relative Neighbouring Localisation algorithm is the relationship between the trends in signal strength and the relative mobility parameters of the neighbouring node on one side and the neighbour's relative location on the other. This relationship is investigated from an analytical perspective in an idealised scenario in this chapter, after which the realistic scenario limitations are examined, along with their impact on localisation procedure. Given the assumed network model and the theoretical basis of the localisation principle, the research problem is redefined as determining single-hop neighbour relative location relying on its mobility parameters and signal strength measurements.

The modular algorithm design, based on the empirical approach employing the Random Forest Regression model, is presented followed by detailed definitions of each step of the algorithm. The basics of the RFR algorithm, as well as the process of selection of that particular regression method, will be presented in the next chapter. In addition, the methodology of the performance evaluation process is also described in the next chapter, along with the simulation and experimental testbed setup. Finally, the next chapter also presents detailed results of the RNL algorithm performance evaluation and proposes a potential application scenario for the proposed localisation algorithm.

Chapter 4

Evaluation

4.1 Introduction

In order to evaluate the performance of the Relative Neighbour Localisation algorithm described in the previous chapter, a series of simulation tests have been undertaken. The simulations need to be calibrated in such a way that they represent reality as close as possible. This is achieved by setting the parameters of the models used in the simulation environment according to the values obtained from the analysis of the measurements obtained using an experimental testbed. This chapter provides the details of the simulation design, the models used, as well as the experimental testbed setup, the tests performed and the analysis of the resulting measurements, used for simulation calibration.

A comparison between a number of regression mechanisms and the process of selection of the estimators used for location angle and distance estimation is also presented in this chapter, followed by the results of the performance analysis of the RNL algorithm and each of its parts, using a series of simulations with different node densities. Finally, an application scenario for distributed sensing coverage control is presented, followed by the performance evaluation of the proposed solution basing on the RNL algorithm and its estimates.

4.2 Methodology

The main method of performance evaluation of the RNL algorithm are simulations, which make large scale testing of the algorithm accuracy in different setups of network and algorithm parameters more feasible in comparison to using an experimental testbed. A simulation environment has been designed using *Python* programming language [101]. In order to calibrate the simulator, a set of real-world experiments have been performed. The data collected in these experiments was used in the process of algorithm design and to derive the parameters of the simulation models. Also, the performance of different regression models was evaluated using the experimental data, in order to choose the one which will be used for the estimators of the RNL. The simulation scenarios aim to replicate the real-world uncertainties introduced by the measurements the RNL algorithm is using as inputs.

4.2.1 Simulation setup

In addition to the notation presented in section 3.2, the notation shown in Table 4.1 will be used for simulation parameters.

Table 4.1 Simulation parameters

Parameter	Description	Unit
R	Dimension of the square area	m
r_c	Communication range	m
(m_{min}, m_{max})	Constant mobility parameters range	s
N	Number of nodes	–
dur	Simulation duration	s

At the beginning of the simulation, N nodes are placed on the $R \times R$ square area by generating each node's x and y coordinates as uniformly distributed variables, i.e. $x, y \sim \mathcal{U}(0, R)$. Node mobility is simulated under the following constraints:

- Nodes are moving at a constant speed of $1 \frac{\text{m}}{\text{s}}$,

- Nodes may not leave the square area,
- Nodes move in a constant direction and constant speed for $m \sim \mathcal{U}(m_{min}, m_{max})$ seconds; m is repeatedly generated after each turn,
- At the beginning of the simulation, and when a turn is due, i.e. m seconds after the previous turn, the direction of movement of each node randomly generated as $\theta \sim \mathcal{U}(-\pi, \pi)$,
- A restriction is placed on the selection of θ: if moving in the selected direction for m_{max} time would place the node outside the square area, the value is discarded and a new one is randomly generated.

The simulation is run as an array of discrete events. This means that every second the locations of all nodes are recalculated, each node updates its neighbourhood list, performs measurement of the strength of the signal from each neighbour, updates the neighbours' mobility parameters, and performs the RNL algorithm.

The strong assumptions regarding the mobility of the nodes in the ad hoc network are made purely for maintaining the simplicity of the simulations. The small variations in speed and direction of movement of nodes from one measurement period to the other can be compensated by averaging, as the algorithm relies on the general information about the speed and direction of movement of the nodes over longer periods of time.

Following are the descriptions of the procedures performed by each node at every simulation step (every second), from the perspective of node i. Neighbourhood update is

done according to the following rule:

$$
\begin{aligned}
d(i,j) \leq r_c \implies & \begin{cases} \text{if} j \in \mathbb{S}_i, \text{ do nothing} \\ \text{if} j \notin \mathbb{S}_i \implies \mathbb{S}_i = \mathbb{S}_i \cup \{j\} \end{cases} \\
d(i,j) > r_c \implies & \begin{cases} \text{if} j \in \mathbb{S}_i \implies \mathbb{S}_i = \mathbb{S}_i \setminus \{j\} \\ \text{if} j \notin \mathbb{S}_i, \text{ do nothing} \end{cases}
\end{aligned}
\tag{4.1}
$$

In the expression above, $\mathbb{S}_i$ represents the set of neighbours of node i, and $d(i,j)$ the true distance between nodes i and j. Signal strength from node $j \in \mathbb{S}_i$ is calculated as:

$$
ss_j^i = p_{T_x} - 10n \log_{10} d(i,j) - X_f, \tag{4.2}
$$

where n is the path loss exponent, p_{T_x} is the transmitting power (in dBm) and X_f is a random variable describing the effects of fading. The derivation of numerical values used in and the justification of this model is done using the data obtained from a real-world experiment (described in section 4.2.2) and is presented in section 4.2.4. The transmitting power p_{T_x} is set to 20 dBm for all nodes. After calculation, signal strength values are averaged with a moving window of n_{avg}. The mobility parameters of each neighbouring node are calculated according to equation (3.11), using the relative speed and direction of movement expressions. The choice, training and use of the estimator component of the RNL localisation algorithm are described in section 4.3.

The basis of the simulation are two classes: `Node` and `Neighbour`. The `Node` class defines each individual node and holds the information about its true location and mobility parameters, as well as the set of `Neighbour` objects. It contains the methods used to move the node, check the neighbourhood and collect the shared estimates for each neighbour in range. The objects of the `Neighbour` class rely on two objects of the `Node` class each – the node whose neighbour set it is part of (e.g. node A), and the node which represents the

neighbouring node itself (e.g. node *B*). Drawing from these two objects, the true relative location and mobility parameters are calculated at each step. The `Neighbour` class also maintains the location estimates (produced by node *A* for the location of node *B*, following the example notation), shared location estimates (of *B* produced by *A*) and the methods for measuring signal strength, running the RNL algorithm and calculating the location estimates based on the shared estimates. The code structure of the simulator is shown in Figure 4.1. The `Simulation` class is responsible for running the simulation using methods for generating, moving and initiating measurements of individual nodes. It maintains the simulation parameters and node list and calculates and stores the localisation errors.

4.2.2 Experimental setup

As previously discussed, the main reason for conducting experimental tests is to quantify the characteristics of a realistic scenario. The environment parameters extracted from the analysis of the experimental data are used to design the simulation models, making the simulation realistic. The real-world experiments are designed to be executable using a reasonable amount of time, space and effort while providing a significantly large dataset for meaningful analysis. This approach means that some changes have to be introduced compared to the scenario investigated throughout this thesis, for practical reasons. One mobile and five stationary nodes are used, placed in different locations around a school hockey field. The nodes communicate among themselves by forming a wireless infrastructureless (ad hoc) network.

The mobile node is broadcasting its direction of movement, and the stationary nodes are measuring the strength of the signal of the moving node every second. Thirty-six different straight line routes are taken by the mobile node, moving at constant speed. The average duration of the routes is around 35 seconds. At different points during the experiment some stationary node wireless network adapters stop producing meaningful measurements, i.e.

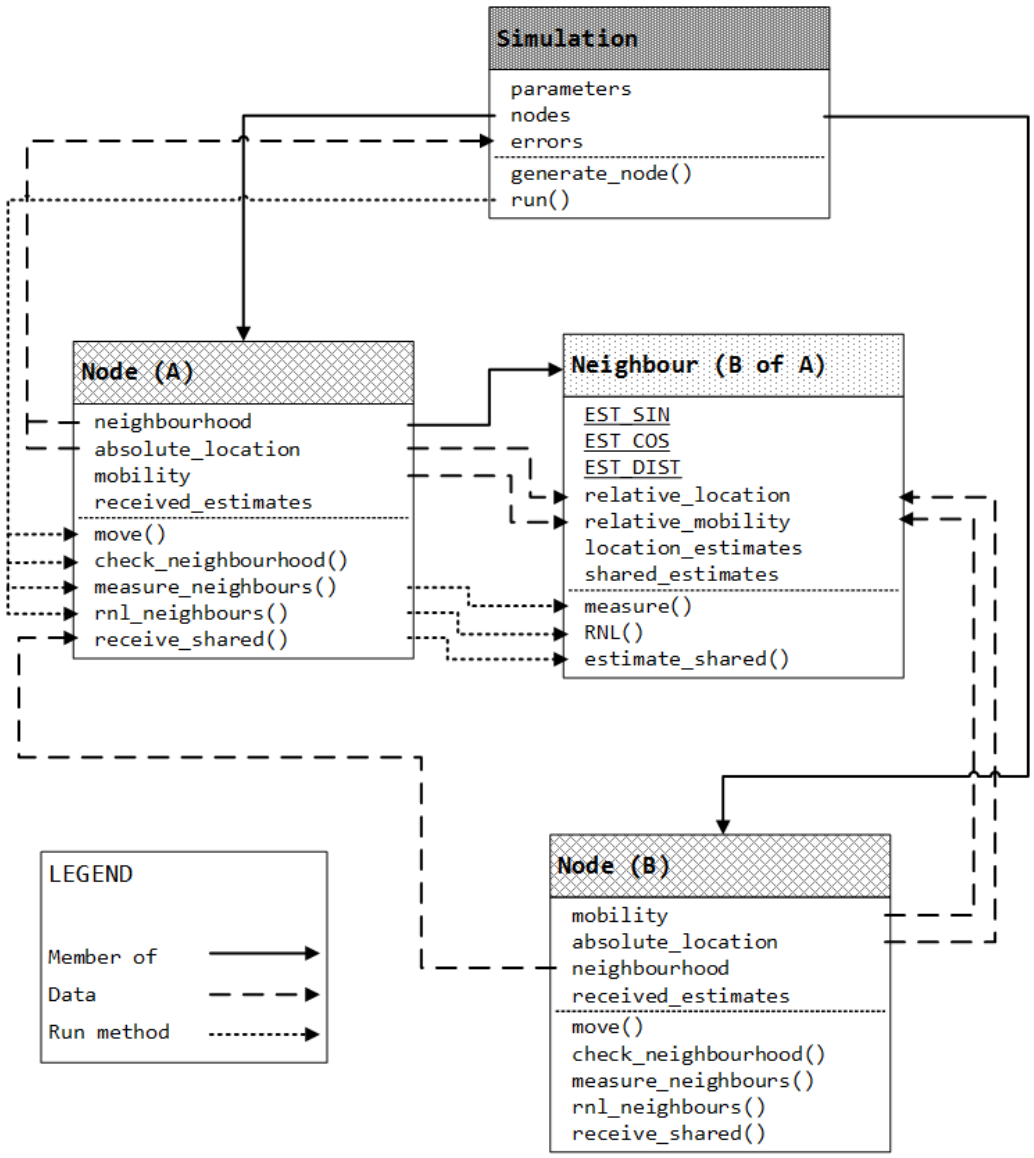


Fig. 4.1 Simulation code diagram

signal strength reading did not change at all. After discarding these measurements, a set of 4340 distinct data points was obtained.

The ground truth for the locations of the stationary nodes and the coordinates of the routes taken by the mobile node is obtained using satellite map data. The satellite image of the testbed location and the positions of the five stationary nodes, as well as four of the thirty-six mobile node routes, are shown in Figure 4.2.

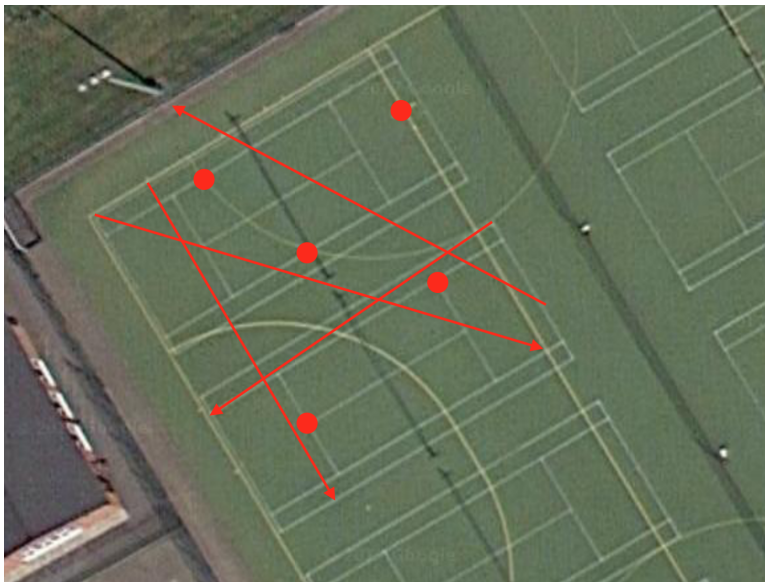


Fig. 4.2 Experimental testbed configuration
Map data: Google, The Geoinformation Group

4.2.3 Hardware

Raspberry Pi model 2 B (RPi) [102] computers are used in the experiment as nodes, both stationary and mobile. The nodes are running a distribution of the Linux operating system - Raspbian, and are equipped with a USB WiFi adapter each, capable of supporting wireless communication according to the IEEE 802.11bgn standards. Nodes are powered by external batteries which allowed for autonomy and full mobility. The information about the direction of movement of the mobile node is obtained from a digital compass connected to the RPi board via *General-Purpose Input/Output* (GPIO) pins. A RPi board with its (GPIO) pins, USB slots and other connectors is shown in figure 4.3.

The mobility of the single mobile node is achieved by mounting in on top of a robot car. The robot car is controlled by connecting its servo controller to the GPIO pins of the mobile node RPi. The servo controller is running the four servo motors, one for each wheel of the car. Additionally, a line tracking module is placed on the bottom of the car, allowing it to autonomously follow a straight line. The full mobile node design is shown in Figure 4.4.

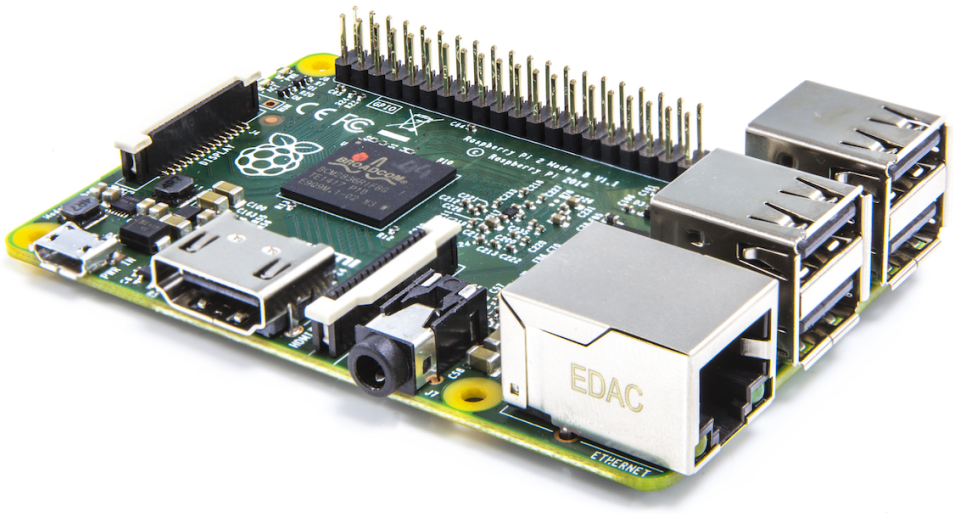


Fig. 4.3 Raspberry Pi model 2 B, adopted from [103]

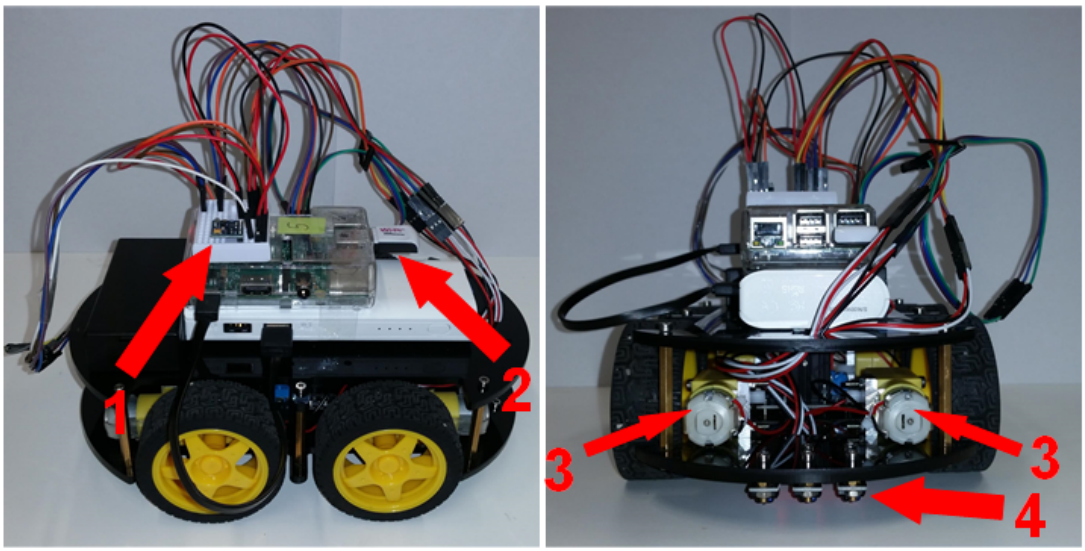


Fig. 4.4 Mobile node design: 1. Digital compass, 2. USB WiFI adapter, 3. Servo motors, 4. Line tracking module

4.2.4 Simulation parameter modelling

In order for the developed simulator to be representative of the real world, the measurements obtained using the experimental testbed are analysed and the simulation parameters are designed so that they reproduce the effects present in the experiment. Two models are

validated using the real world measurements – the digital compass bearing readings and the propagation model, i.e. the effects of fading on signal strength relative to the distance between the sender and receiver.

Bearing readings

The bearing angle readings which were acquired from the digital compass module used on the mobile node use the direction of the East as the zero reference. As in standard bearing angle reading, the angle values are increasing clockwise. For practical purposes, the ground truth of the stationary nodes' locations and the mobile node routes' coordinates are calculated in a coordinate system which is aligned to the edges of the hockey field, as shown in Figure 4.5. The x and y axes are intentionally placed this way so that the angles calculated using this coordinate system are also increasing clockwise.

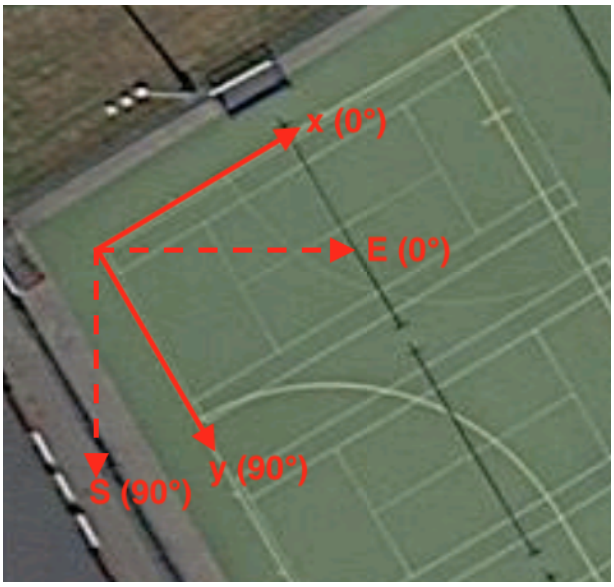


Fig. 4.5 The local and global coordinate systems
Map data: Google, The Geoinformation Group

Even though the mobile node is moving in a straight direction, the compass angle readings expectedly fluctuate over time, mostly due to the slight turns which the robot car needs to perform when it detects that it has steered off the line it is supposed to be tracking, as well

as the terrain not being ideally flat. These variations in the reported bearing angle will be addressed later. For now, the average values across single routes are calculated and compared to the ones computed using the local coordinate system and the start and end points of each route. The offset between the local and global reference system-based angle values are shown in Figure 4.6, where each point on the graph shows one route of the mobile node. The linear approximation of this relationship is described as $y = x - 0.71$, which fits the coordinate system offset described above.

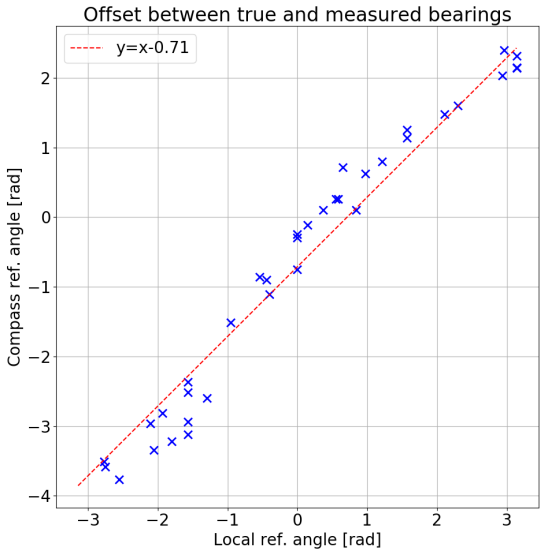


Fig. 4.6 The offset between the true and measured direction angles

Due to the imperfect calibration and the limited sensitivity and accuracy of the sensor certain errors in reported bearing angles are present, which can also be noted from Figure 4.6. The set of differences between all individual bearing readings and true values (compensated for the coordinate system offset) exhibit a non-analytical distribution, not favourable for modelling. However, if a more precise and more accurate sensor was used, as well as optimal calibration and better mobility control, it could be assumed that the mean values of movement direction angle correspond to the true bearing. In this case, the errors are investigated on an individual route level, and their distribution is shown in Figure 4.7. Disregarding the small tail on the left side of the distribution, these errors can be modelled using a Gaussian

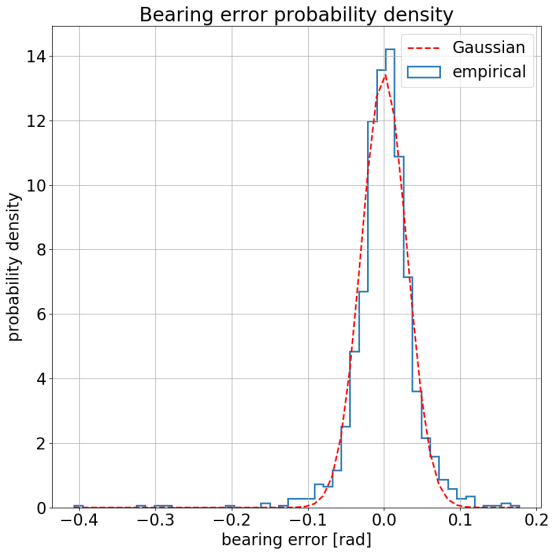


Fig. 4.7 Probability density of bearing errors

distribution with a mean of 0 and variance of $8 \times 10^{-4}\,\text{rad}^2$. This model will be used in the simulation by adding a Gaussian random variable to the true bearing of moving nodes. The bearing readings can now be simulated as:

$$\theta = \theta_{true} + X_\theta, \;\; X_\theta \sim \mathcal{N}\left(0, 8 \times 10^{-4}\right), \tag{4.3}$$

where θ_{true} is the true direction of movement and all values are in radians.

Propagation model

The second, more important model which will arise from the experimental data analysis is the one regarding radio signal propagation. As previously discussed, multipath propagation causes signal fading effects, which is the biggest source of uncertainty in the RNL localisation approach. In order to design the simulation propagation model, a simple path loss model is first adopted, which allows the baseline expected signal strength to be calculated as a function of distance. Path loss represents the difference between the transmitted and received

power. This path loss model (in decibels) is described as:

$$PL = 10n\log_{10} d + C, \tag{4.4}$$

where n is the path loss exponent, d is the distance between the transmitting and receiving nodes, and C is a constant describing the effects of fading and all other losses combined. From all the measurements obtained from the experimental testbed (4340 of them), pairs of true distance and measured signal strength are formed. Graphical representation of the *distance to signal strength mapping* is shown in Figure 4.8. Along with all the measurements, the fitted model corresponding to the equation (4.4), and transmit power of 100 mW, which is equivalent to 20 dBm, is shown.

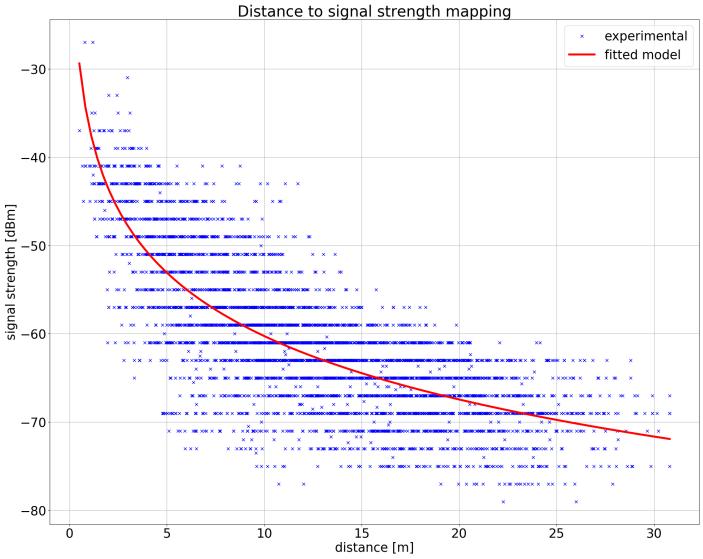


Fig. 4.8 Experimental *distance – signal strength* pairs

The fitted path loss equation produced the following values for the path loss parameters: $n = 2.39$ and $C = 56.41$ dB. Looking at the experimental signal strength values, it can be concluded that instead of using a constant to describe the effects of fading, a better choice

would be a random variable. This approach transforms equation (4.4) into:

$$PL = 10n\log_{10} d + X_f. \tag{4.5}$$

The previously obtained fitted model can be used to design the fading random variable, by analysing the distribution of the errors made by the deterministic model from (4.4). The measurements are grouped based on distance, using 0.5 m wide bins and errors are calculated with reference to the deterministic model-based signal strength calculated for the centre-point of each bin. These errors in the deterministic model represent the varying effects of fading, the distribution of which is shown in Figure 4.9.

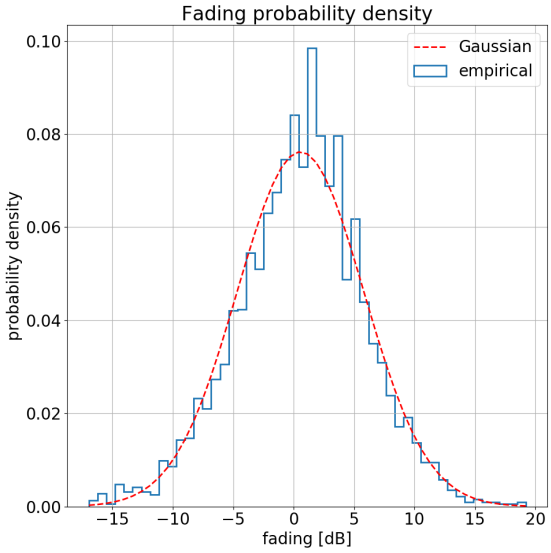


Fig. 4.9 Probability density of fading

The effects of fading compared to the deterministic signal strength computation can be modelled using a Gaussian distribution with mean 0.57 dB and variance 27.43 dB^2.[1] The relation between the probabilistic X_f from equation (4.5) and the deterministic C from

[1] The author acknowledges the absurdity of squaring decibels, as well as analysing logarithmic values in a linear way. However, the presented analysis is aimed at describing the numerical effects fading has on signal strength values in decibels in order for these effects to be accurately modelled in a simulation environment.

equation (4.4) can be expressed as:

$$\begin{aligned} X_f &= C + \mathcal{N}(0.57, 27.43) \\ X_f &\sim \mathcal{N}(C + 0.57, 27.43)\,, \end{aligned} \tag{4.6}$$

which allows the final path loss simulation model to be expressed as (in decibels):

$$PL = 10n\log_{10} d + X_f,\;\; X_f \sim \mathcal{N}(56.98, 27.43)\,. \tag{4.7}$$

Observing the experimental dataset it can be concluded that signal strength measurements are reported as integers and mostly odd numbers, due to the characteristics of the wireless adapters used in the experiment. This would not be the case if a wireless adapter capable of reporting signal strength measurements with greater granularity was used. Therefore, the simulated signal strength measurements will not be rounded to whole numbers and no bias towards odd numbers will be included in the model. A conclusion about the communication range in the experimental scenario can also be drawn. One way of looking at it is that the maximum communication distance is around 30 m. The other one is related to the detectable levels of signal strength, the minimum of which is 79 dB. In line with the previous assumption about better performance wireless adapters being used, as well as for practical reasons, in simulation scenarios, the distance between two nodes will be the limiting factor for communication range.

The results of simulating signal strength measurements using the path loss model from (4.7) on the set of experimental distances are shown in Figure 4.10. These results are similar to the experimentally obtained ones, shown in Figure 4.8.

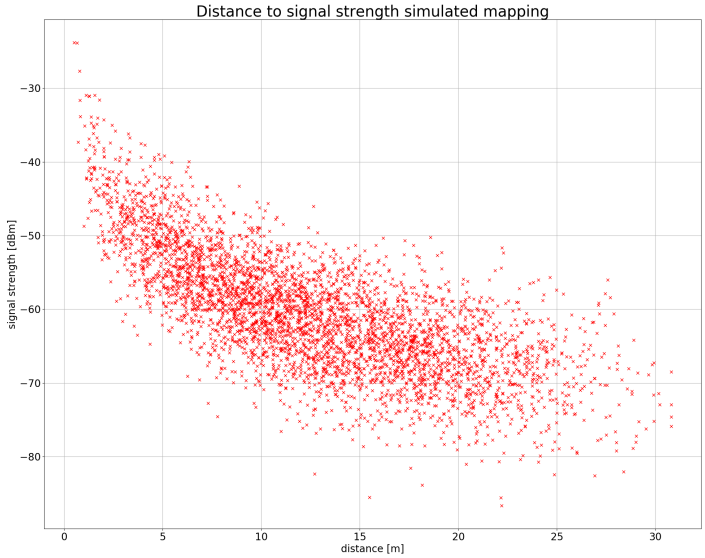


Fig. 4.10 Simulated *distance – signal strength* pairs

4.3 Regression Method Selection

This section will cover the process of selection of the regression algorithm used for location angle and distance estimation, as described in sections 3.3.1 and 3.4. The choice is made based on the performance of each technique for the task at hand. In the case of RNL algorithm the dataset, as previously described, is composed of signal strength measurements, bearing angles and speeds as inputs, and location angles as outputs. Given the complexity of the relationship between these parameters in a realistic scenario, an empirical approach is chosen in order to produce the angle and distance estimates from available measurements. Based on an offline learning phase, the estimator creates a set of rules which map inputs to outputs. These rules are later applied in the online phase during algorithm use on any new inputs to produce the output estimates.

4.3.1 Regression methods

Regression analysis is used for building models which estimate the relationship between the independent variables and the dependent variable. In general, the analysis is started by

analysing a dataset which contains input points (a vector of independent variable values) with known values of output points (one of the dependent variables). This is referred to as *the offline phase* or *training of the model*. Depending on the regression method used, a set of rules is constructed based on the relationships the input – output pairs exhibit, describing in which way the dependent variable depends on the independent ones. These rules can then be applied to any new vector of input values to produce an estimate of the output value corresponding to it. The application of the set of rules as an estimator is referred to as *the online phase*.

In the case of the estimators used in the RNL algorithm, the independent variables are the signal strength measurements and the mobility parameters of the target node, and the dependent variable is one of the three parameters the values of which are estimated: *sine and cosine of the location angle*, and the *distance* of the target node. The values of the hyperparameters of each of the estimators have been chosen based on the estimation results achieved, choosing the set of values which provided the best estimates.

Decision Tree Regression (DTR)

Decision trees methods work by dividing the space of all possible values of the input vector into regions [104], based on individual parameter's values. During the training phase, each of the regions is assigned an output value, which is the mean output of all input points belonging to the region. The splitting of the set of all training data points into regions, or *leaves*, is performed in a greedy, top-down manner. A split is defined by the input vector variable it concerns and its value, based on which the split is made. Data points with the value of that variable below the split value are assigned into one region, while the ones with the value of the variable greater than the splitting value are assigned to the other. The variable and its value which will be used as the splitting criterion are chosen in a way which will result in minimising the sum of variances of the two resulting regions of data points, i.e. the splitting

is done by data point "similarity". Multiple criteria for stopping the splitting procedure can be used, some of which are the depth of the tree, the minimum number of data point per leaf, the minimum number of data points per leaf for it to be split further, or the maximum number of leaves. An example of data split into regions based on different variable values is shown in Figure 4.11.

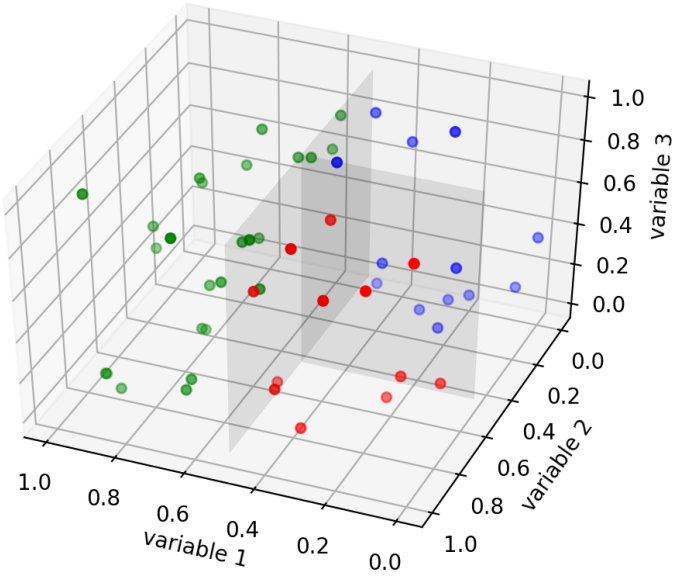


Fig. 4.11 Illustration of a dataset split into regions

The following values have been used for hyperparameters:

- maximum tree depth – No limit
- minimum samples per leaf – 2
- minimum samples for split – 2
- maximum number of features to be considered for a split – No limit
- maximum number of leaf nodes – No limit

Random Forest Regression (RFR)

This regression method represents an example of ensemble learning or a combination of multiple regressors [105]. It uses multiple DTR estimators, by training each on a randomly

chosen subset of the original training dataset. Furthermore, at each split, a random subset of the input vector variables is chosen for consideration. This way, the phenomenon known as *overfitting* the model is avoided. Overfitting happens when the model is fit so closely to the training dataset that it begins to model the noise in the data, rather than just the principal relationship between the independent and dependent variable. As a result, the model is incapable of producing good estimates on anything other than the training dataset. The output of the RFR estimate is calculated as the mean output of each of its DTR estimators.

The following values have been used for hyperparameters:

- number of estimators – 10
- maximum tree depth – No limit
- minimum samples per leaf – 2
- minimum samples for split – 2
- maximum number of features to be considered for a split – No limit
- maximum number of leaf nodes – No limit

K-Nearest Neighbours (KNN)

Given an input with unknown output, which is to be estimated, the nearest neighbour algorithm [106] searches for the data points from the training set which are closest, according to the chosen metric (like Euclidean or Manhattan distances). The number k defines how many closest neighbours are to be considered. Once the k nearest neighbours have been found in the training data set, the output is estimated based on their known outputs. Weighing can be applied so that the contribution of individual neighbours depends on their distance to the new input.

The following values have been used for hyperparameters:

- number of neighbours – 5
- distance metric – Euclidean

Support Vector Regression (SVR)

This regression method is based on the support vector machine [107]. This algorithm was originally intended for classification problems, where the outputs can only take discrete values from a predefined set, rather than any of the infinite real values as in regression problems. The SVR algorithm works by trying to fit a hyperplane through the data points, allowing a certain margin around the hyperplane within which the data points can be located. In order to reduce the complexity of the potential solution, non-linear transformations of independent variables can be performed before the fitting procedure. The cost function being minimised is related to the points lying outside the margin, thus producing the best fit to describe the relationship between the independent and dependent variables.

The following values have been used for hyperparameters:

- kernel type – radial basis function
- stopping criterion tolerance – 0.001
- error penalty parameter – 1

4.3.2 Estimator performance comparison

The estimator performance is first evaluated on the experimental dataset used in section 4.2.2. All the data points are put together, their order randomised and they are split into ten equal groups. In each of the ten tests, one of the groups is used for testing, and the other nine for training of the estimator. The results shown below are cumulative for all of the ten test runs. As previously described in chapter 3, three separate estimators are used for the sine and cosine values of the location angle and for distance. Location angle and the relative location are then calculated from these estimates. Symmetrical angle estimates are calculated using the raw estimate and the received direction angle. Because symmetrical angle handling is not implemented in this test, the smaller of the two errors (raw and symmetrical) is always used,

for both the location angle and the location estimates, making these results better than the ones that can be expected in the real application of the estimators. Tables 4.2, 4.3 and 4.4 show the angle, distance and location estimate accuracy for three different values of n_{prev}: $3, 5,$ and 7, respectively, for all of the tested regression mechanisms.

Table 4.2 Estimator accuracy for $n_{prev} = 3$ on experimental dataset

method	estimate	error			
		mean	percentile		
			25	50	75
RFR	angle	44.76°	13.02°	31.23°	63.22°
	distance	2.55 m	0.95 m	1.97 m	3.59 m
	location	8.94 m	3.58 m	6.83 m	12.15 m
SVR	angle	50.92°	15.40°	38.46°	75.11°
	distance	2.72 m	0.98 m	2.13 m	3.80 m
	location	10.11 m	4.00 m	8.15 m	13.67 m
KNN	angle	44.98°	11.97°	30.68°	64.40°
	distance	2.49 m	0.82 m	1.82 m	3.48 m
	location	8.92 m	3.17 m	6.57 m	12.29 m
DTR	angle	52.74°	9.72°	37.65°	83.23°
	distance	3.16 m	0.79 m	2.22 m	4.48 m
	location	10.40 m	3.55 m	8.02 m	14.80 m

Observing the values presented in these three tables, it can be concluded that increasing n_{prev} in general increases accuracy. This is not surprising, but this increase of accuracy comes at the price of location estimate not being available for a longer time after a new neighbour appears in range or an existing one makes a turn. This is why the choice of $n_{prev} = 5$ is made for further testing and implementation, as a balance between accuracy and estimate availability. Regarding individual estimators performance, the RFR and KNN estimators achieve higher accuracy than the other methods, and their results are close to each other.

Three issues exist with the experimental dataset – the first one is that it does not include varying relative speed as there is only one moving node, the second one is that its limited size (around 3500 data points, depending on individual routes), and the last one is its low variety,

Table 4.3 Estimator accuracy for $n_{prev} = 5$ on experimental dataset

method	estimate	error			
		mean	percentile		
			25	50	75
RFR	angle	41.54°	11.72°	27.68°	57.96°
	distance	2.43 m	0.90 m	1.91 m	3.42 m
	location	8.38 m	3.28 m	6.15 m	11.11 m
SVR	angle	48.76°	14.33°	34.78°	72.62°
	distance	2.60 m	0.90 m	1.97 m	3.63 m
	location	9.58 m	3.71 m	7.42 m	12.96 m
KNN	angle	39.03°	10.31°	26.47°	54.00°
	distance	2.23 m	0.72 m	1.63 m	3.11 m
	location	7.78 m	2.68 m	5.69 m	10.54 m
DTR	angle	48.75°	6.98°	31.66°	76.92°
	distance	2.85 m	0.68 m	1.91 m	4.11 m
	location	9.58 m	2.99 m	6.94 m	13.52 m

Table 4.4 Estimator accuracy for $n_{prev} = 7$ on experimental dataset

method	estimate	error			
		mean	percentile		
			25	50	75
RFR	angle	35.42°	9.57°	23.16°	46.47°
	distance	2.33 m	0.84 m	1.79 m	3.29 m
	location	7.31 m	2.81 m	5.30 m	9.36 m
SVR	angle	45.45°	12.94°	30.69°	67.96°
	distance	2.50 m	0.83 m	1.87 m	3.52 m
	location	8.93 m	3.42 m	6.64 m	12.11 m
KNN	angle	33.67°	9.13°	22.35°	44.18°
	distance	2.03 m	0.62 m	1.47 m	2.84 m
	location	6.86 m	2.40 m	4.81 m	8.98 m
DTR	angle	42.66°	5.02°	23.66°	66.49°
	distance	2.69 m	0.63 m	1.63 m	3.88 m
	location	8.60 m	2.35 m	5.83 m	12.36 m

as multiple data points belong to the same runs, and the number of different scenarios is low, which is caused by the nature of the experiment.

In order to test the regressors on a larger scale and include relative node speed as a variable, the evaluation procedure is performed using datasets obtained through simulation as well. Two separate datasets are generated, one for training and testing each. The simulation setup used to generate the datasets is the same as the one described in section 4.2.1. Simulation parameters for regression techniques performance evaluation datasets are shown in Table 4.5.

Table 4.5 Regression test simulation parameter values

Parameter	Value	Unit
R	100	m
r_c	30	m
(m_{min}, m_{max})	$(15, 20)$	s
N	50	-
dur	50	s

At this point, it is worth reiterating that the experimental dataset has its shortcomings and that simulation dataset-based results are more representative, as well as the fact that in the estimator tests no handling of symmetrical angles has been applied, and the better of the two estimates (raw and symmetrical) is always used, for both the location angle and the location estimates. Table 4.6 shows the same set of estimators on a much larger simulation dataset (approximately 30000 data points). After including relative speed as a relevant input as well as having a more diverse training set, RFR estimators shows the best location estimate accuracy.

In addition, the ratio of the training complexity of the two methods with best accuracy results, SVR and RFR, is $x^2 : \log x$. In a hypothetical application scenario in which measurements are taken in order to estimate the signal propagation model parameters, based on which a simulation dataset can be created and the estimator trained, all at the time of network

Table 4.6 Estimator accuracy for $n_{prev} = 5$ on simulation dataset

method	estimate	error			
		mean	percentile		
			25	50	75
RFR	angle	44.85°	14.71°	34.94°	66.45°
	distance	2.82 m	1.05 m	2.29 m	4.07 m
	location	13.76 m	4.85 m	10.14 m	20.27 m
SVR	angle	45.52°	15.09°	35.79°	67.17°
	distance	3.07 m	1.22 m	2.53 m	4.43 m
	location	13.95 m	5.31 m	10.46 m	20.10 m
KNN	angle	46.13°	15.91°	36.80°	68.31°
	distance	3.04 m	1.15 m	2.47 m	4.37 m
	location	14.04 m	5.27 m	10.63 m	20.44 m
DTR	angle	46.86°	16.03°	37.17°	69.07°
	distance	3.82 m	1.32 m	3.01 m	5.52 m
	location	14.70 m	5.73 m	11.49 m	21.16 m

deployment, or immediately before, this difference can result in big training time difference, severely hurting the efficiency of the SVR estimator.

4.4 Localisation Results

The performance of the RNL algorithm is evaluated using a series of simulations based on the simulation setup described above. The simulations consist of a number of nodes moving around a square area for a period of time. All nodes repeatedly perform the RNL algorithm on all of the neighbours within their communication range, and all of their estimates are recorded throughout the simulation. The estimates considered can be grouped into *location angle*, *distance* and *relative two-dimensional location*. Given the flow of the RNL algorithm, where the two-dimensional location estimates are calculated from the location angle and the distance estimates, these two basic estimates can be observed as standalone outputs of the RNL algorithm, or more precisely, the parts dedicated to their estimation. Only

with estimate sharing does the two-dimensional relative location become necessary for the algorithm operation.

Having this in mind, the estimates from each of the categories are examined at different steps of the RNL algorithm, thus providing insight into the performance of the individual algorithm blocks, as well as the algorithm as a whole. In line with the distributed nature of the algorithm, where all estimates are produced relative to the node which is performing localisation using the RNL algorithm, relative positions of each pair of neighbouring nodes are calculated at each step of the simulation, and are used to evaluate individual estimates in each of the categories. The estimates are grouped and analysed at the level of the whole simulation. The simulation parameters are shown in Table 4.7. Detailed results for each of the location angle, distance and two-dimensional location estimates are presented below.

Table 4.7 Simulation parameters

Parameter	Value	Unit
R	100	m
r_c	30	m
(m_{min}, m_{max})	$(15, 20)$	s
N	$\{20, 50, 100\}$	-
dur	50	s

4.4.1 Location angle estimates

The errors in angle estimates are evaluated by calculating the smaller angle between the lines passing through the root node at the *true location angle* and the *estimated location angle*. This way, it is assured that no error greater than 180° is possible, which is in line with the periodic nature of the angle quantity. The errors are calculated using the following equation:

$$\delta_{ang} = \arctan\left(\frac{\sin(\alpha - \alpha_{est})}{\cos(\alpha - \alpha_{est})}\right), \tag{4.8}$$

where α is the true location angle and α_{est} represents a location angle estimate. All errors in this section will be presented converted to degrees.

Considering first the location angle estimation block on its own, there exist four location angle estimates of interest:

γ_R the raw RFR angle estimate – this is the value calculated based on the outputs of the RFR sine and cosine estimation blocks,

$\hat{\alpha}_1, \hat{\alpha}_2$ the two ambiguous angle estimates – produced as the result of symmetrical angle processing, before ambiguity resolution is possible,

$\hat{\alpha}$ the unique angle estimate – produced as the result of symmetrical angle processing, after ambiguity has been resolved.

The results of these estimate errors vary only slightly in different simulation runs with a different number of nodes in the network. This is due to a lower number of nodes resulting in a lower number of overall estimates, and with an increase in the number of nodes (and total estimates made), the results start to converge. On the other hand, the fact these differences are small is expected, as they represent only the standalone performance of the angle estimation block of the RNL algorithm, without any impact of estimate sharing on them. Table 4.8 shows the estimates errors of the four estimates produced by the angle estimation block (in a simulation with $N = 50$).

Table 4.8 Errors of individually produced location angle estimates

estimate	error			
	mean	percentile		
		25	50	75
γ_R	67.71°	25.95°	57.99°	103.03°
$\hat{\alpha}_1$	69.49°	26.14°	60.62°	106.86°
$\hat{\alpha}_2$	69.43°	27.07°	60.11°	105.29°
$\hat{\alpha}$	56.79°	18.58°	43.87°	85.07°

Without any processing, the median raw RFR estimate (γ_R) error is around 58°. The two ambiguous angle estimates ($\hat{\alpha}_{1,2}$) perform slightly worse, at around 60°. This deterioration is due to the fact that these two ambiguous estimates share among themselves the combined inaccuracy of the raw and symmetrical RFR estimates, and given the fact that the difference between these two estimates' errors and the raw RFR estimate error is small (but greater), it can be concluded that the raw RFR estimates have a slight bias to the true location angle, compared to the symmetrical one. On the other hand, once the ambiguity is resolvable, and the symmetrical angle processing block is able to produce a unique estimate, the accuracy increases significantly, resulting in a median error of under 45°. Comparing the mean and the median of the estimates errors shows that across all estimates the mean is considerably greater than the median, suggesting that very large errors can be considered outliers.

Estimate sharing is based on two-dimensional relative location estimates ($\hat{x}_{sh}, \hat{y}_{sh}$), so the shared angle estimates are calculated from the relative location produced as the output of the estimate sharing block as:

$$\hat{\alpha}_{sh} = \arctan\left(\frac{\hat{y}_{sh}}{\hat{x}_{sh}}\right), \tag{4.9}$$

while their errors are calculated using equation (4.8). In the shared estimates analysis, the following three location angle estimate values are of interest:

$\hat{\alpha}_{sh}$ the location angle estimate calculated from the shared location estimate,

$\hat{\alpha}_{oic}$ the shared location angle estimate when (one of) the locally produced estimate(s) (own estimates) is part of the cluster chosen in the estimate sharing procedure,

$\hat{\alpha}_{nic}$ the shared location angle estimate when no locally produced estimate is part of the cluster chosen in the estimate sharing procedure.

The number of nodes in the network directly impacts node density, and the expected number of neighbouring nodes with which estimates can be shared. Tables 4.9, 4.10 and 4.11 show

the errors in own and the three types of shared location angle estimates, for different number of nodes in the network.

Table 4.9 Errors of shared and locally produced location angle estimates ($N = 20$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\alpha}$	61.95°	21.82°	47.65°	97.65°
$\hat{\alpha}_{sh}$	61.03°	18.90°	47.31°	96.71°
$\hat{\alpha}_{oic}$	57.18°	17.70°	42.75°	88.31°
$\hat{\alpha}_{nic}$	68.14°	22.35°	54.22°	109.67°

Table 4.10 Errors of shared and locally produced location angle estimates ($N = 50$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\alpha}$	56.79°	18.58°	43.87°	85.07°
$\hat{\alpha}_{sh}$	56.06°	18.90°	42.62°	82.89°
$\hat{\alpha}_{oic}$	49.85°	15.90°	36.34°	71.72°
$\hat{\alpha}_{nic}$	61.50°	22.19°	48.73°	92.17°

Table 4.11 Errors of shared and locally produced location angle estimates ($N = 100$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\alpha}$	58.84°	18.43°	44.57°	90.86°
$\hat{\alpha}_{sh}$	55.14°	17.70°	40.72°	82.11°
$\hat{\alpha}_{oic}$	47.11°	14.26°	32.79°	66.65°
$\hat{\alpha}_{nic}$	59.26°	19.90°	45.41°	89.51°

In general, angle estimates from shared location estimates show smaller errors that the processed unique angle estimates. A greater benefit of estimate sharing is achieved if the shared location estimates are discriminated based on the presence of on own estimate in the

cluster of chosen estimates in the estimate sharing step of the RNL algorithm. This is also where the impact of the node density can be noticed. The median estimate error is reduced by an increasing amount as the number of nodes in the network (and consequentially, the number of neighbours) increases. The error reduction is around 5° for $N = 20$, 8° for $N = 50$ and 12° for $N = 100$. Comparing the errors of the initial, raw RFR location angle estimate and the one obtained by the estimate sharing procedure (with an own estimate contributing to the shared one), shows the overall benefit of the symmetrical angle and estimate sharing steps of the RNL algorithm. The relative improvement of median errors for different number of nodes is shown in Table 4.12. Figure 4.12 shows the step-by-step improvement of the angle estimates, from raw RFR estimates, over estimates processed for ambiguity, and finally shared estimates.

Table 4.12 Median angle estimate error reduction: raw to shared

estimate	median error		
	$N = 20$	$N = 50$	$N = 100$
γ_R	55.38°	57.99°	59.44°
$\hat{\alpha}_{oic}$	42.75°	36.33°	32.79°
rel. imp.	0.77	0.62	0.55

It can be concluded that both the symmetrical angle handling and the estimate sharing are beneficial to the angle estimation part of the RNL algorithm, reducing the initial estimate error by as much as 45%.

4.4.2 Distance estimates

The distance estimate errors are evaluated by calculating the absolute difference between the *true distance* and the *estimated distance* to the target node. This means that distance estimate errors are expressed as positive numbers and are calculated as:

$$\delta_d = |d - d_{est}|, \tag{4.10}$$

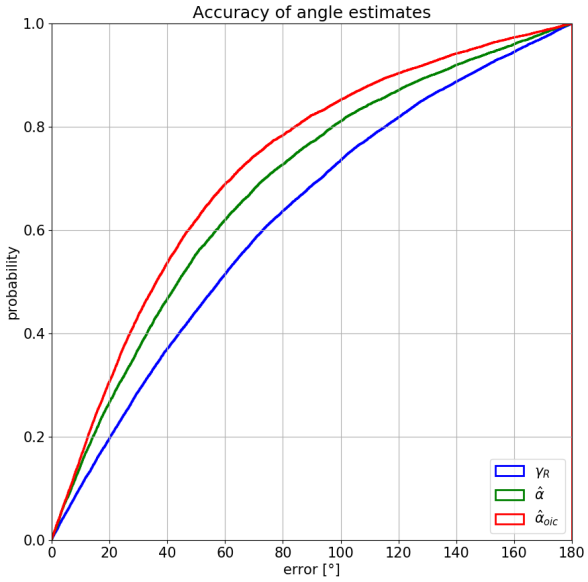


Fig. 4.12 Improvement of angle estimates

where d_{est} represents the distance estimate. Unlike angle estimates, the RNL locally (without the estimate sharing block) produces only one distance estimate, using the RFR distance estimation block. After sharing the two-dimensional relative location estimates $(\hat{x}_{sh}, \hat{y}_{sh})$ with the nodes in range, and calculating the final shared location estimate, the distance component of the shared estimate can be calculated as:

$$\hat{d}_{sh} = \sqrt{\hat{x}_{sh}^2 + \hat{y}_{sh}^2}, \tag{4.11}$$

and the error of the shared distance estimate is calculated using equation (4.10). For the distance estimate analysis, the following four distance estimates are of interest:

$\hat{d}$ the RFR distance estimate,

$\hat{d}_{sh}$ the distance estimate calculated from the shared location estimate,

$\hat{d}_{oic}$ the shared distance estimate when (one of) the locally produced estimate(s) (own estimates) is part of the cluster chosen in the estimate sharing procedure,

$\hat{d}_{nic}$ the shared distance estimate when no locally produced estimate is part of the cluster chosen in the estimate sharing procedure.

The number of nodes in the network defines the node density and the expected number of received estimates from neighbours during the estimate sharing procedure. Tables 4.13, 4.14 and 4.15 show the errors in own and the different types of shared estimates for $N = 20$, $N = 50$, $N = 100$ nodes in the network respectively.

Table 4.13 Errors of shared and locally produced distance estimates ($N = 20$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{d}$	2.92 m	1.00 m	2.17 m	4.09 m
$\hat{d}_{sh}$	6.59 m	2.22 m	5.11 m	9.57 m
$\hat{d}_{oic}$	5.92 m	1.94 m	4.48 m	8.46 m
$\hat{d}_{nic}$	7.82 m	2.97 m	6.22 m	11.25 m

Table 4.14 Errors of shared and locally produced distance estimates ($N = 50$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{d}$	3.02 m	1.01 m	2.32 m	4.24 m
$\hat{d}_{sh}$	5.99 m	1.97 m	4.37 m	8.43 m
$\hat{d}_{oic}$	4.62 m	1.55 m	3.41 m	6.43 m
$\hat{d}_{nic}$	7.19 m	2.56 m	5.54 m	10.23 m

The main difference between the distance and the location angle estimations is the fact that the estimate sharing procedure is actually reducing the accuracy of the distance estimates. This can be explained in two ways. The first one is the fact that the initial, locally made, already exhibit a relatively high level of accuracy, with the median error always below 2.5 m. Again, as is the case with the angle estimates, the mean error is greater than the median

Table 4.15 Errors of shared and locally produced distance estimates ($N = 100$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{d}$	3.05 m	1.02 m	2.32 m	4.27 m
$\hat{d}_{sh}$	5.56 m	1.75 m	3.99 m	7.75 m
$\hat{d}_{oic}$	3.88 m	1.26 m	2.87 m	5.40 m
$\hat{d}_{nic}$	6.42 m	2.14 m	4.80 m	9.15 m

error, suggesting there exist large error outliers. The second reason why estimate sharing has an adverse effect on the accuracy of distance estimates is that the estimates received from neighbouring nodes in the process of sharing estimates are computed based on their location angle and distance estimates. And while the neighbours' distance estimates are also relatively accurate, the precision of the two-dimensional relative location estimate is diluted by the error in the location angle, resulting in the distance estimate component of the shared estimate to be less accurate than the locally produced one.

On the other hand, the shared distance estimate errors behave in a similar way as the shared location angle ones. With the increase in the number of neighbours which can share their estimates, the number of received estimates increases, and the accuracy of the shared distance estimate increases. Again, when the locally produced estimate is part of the cluster chosen by the estimate sharing block, the shared distance estimate is more accurate in comparison to when this is not the case. With the increase of the number of nodes in the network, the locally produced distance estimate exhibits slight increase across all percentiles, which can be attributed to the estimate error converging as the overall number of estimates produced in the simulation increases.

4.4.3 Location estimates

The two-dimensional location estimates are evaluated by calculating the Euclidean distance between the *true relative location* (x,y) and the *estimated relative location* (x_{est},y_{est}) of the target node, according to the following equation:

$$\delta_{loc} = \sqrt{(x_{est}-x)^2+(y_{est}-y)^2}. \tag{4.12}$$

The local location estimates of the target node are calculated from the location angle and distance estimates as:

$$\begin{aligned} \hat{x} &= \hat{d}\cos(\hat{\alpha}) \\ \hat{y} &= \hat{d}\sin(\hat{\alpha}), \end{aligned} \tag{4.13}$$

and there may exist one unique or two ambiguous location estimates, depending on the target node location angle estimate ambiguity. In the process of estimate sharing the unique or both the ambiguous location estimates are share with the neighbours, and depending on the ambiguity of the estimate of the neighbour location estimate there may be one, two or four total locations which participate in the clustering procedure. If the own and shared location estimates are defined as:

- $\hat{\mathbb{L}}$ the locally produced location estimate; from $\hat{\alpha}$ and $\hat{d}$,
- $\hat{\mathbb{L}}_{sh}$ the location estimate calculated from the shared estimates,
- $\hat{\mathbb{L}}_{oic}$ the shared location estimate when (one of) the locally produced estimate(s) (own estimates) is part of the cluster chosen in the estimate sharing procedure,
- $\hat{\mathbb{L}}_{nic}$ the shared location estimate when no locally produced estimate is part of the cluster chosen in the estimate sharing procedure.

the general expression for the location estimate error (equation (4.12)) can be rewritten as:

$$\delta_{loc} = ||\mathbb{L} - \mathbb{L}_{est}||, \tag{4.14}$$

where $\mathbb{L} = (x, y)$ is the true relative location of the target node. For the purpose of comparing own and shared location estimates, only the unique locally produced location estimates are considered, while both unique and ambiguous location estimates participate in the process of shared location estimate calculation.

Before presenting the location estimate errors, it is worth reiterating the conclusions of the previous two sections, where the estimate sharing procedure was shown to significantly increase the accuracy of the location angle estimates, and reduce the accuracy of the distance estimate. The errors of location estimates for $N = 20$, $N = 50$ and $N = 100$ nodes in the network are shown in Tables 4.16, 4.17 and 4.18, respectively.

Table 4.16 Errors of shared and locally produced location estimates ($N = 20$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\mathbb{L}}$	17.00 m	6.54 m	13.58 m	25.12 m
$\hat{\mathbb{L}}_{sh}$	17.01 m	7.27 m	14.21 m	24.12 m
$\hat{\mathbb{L}}_{oic}$	14.95 m	6.46 m	12.82 m	20.87 m
$\hat{\mathbb{L}}_{nic}$	21.05 m	9.25 m	18.61 m	40.00 m

The shared relative location estimates produce smaller errors than the locally produced ones. This information, combined with the behaviour of the location angle and distance estimates with and without estimate sharing leads to the conclusion that the main source of the two-dimensional location estimate error is the location angle part of it. Through the process of sharing, regardless of the deterioration of the distance estimate, the location angle component ends up being improved significantly enough for the shared estimate to be more accurate than the locally produced one. The effects of shared location estimates are

Table 4.17 Errors of shared and locally produced location estimates ($N = 50$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\mathbb{L}}$	17.06 m	5.63 m	12.76 m	26.39 m
$\hat{\mathbb{L}}_{sh}$	18.19 m	7.98 m	15.56 m	26.24 m
$\hat{\mathbb{L}}_{oic}$	14.83 m	5.96 m	11.72 m	21.01 m
$\hat{\mathbb{L}}_{nic}$	20.96 m	10.73 m	19.20 m	29.52 m

Table 4.18 Errors of shared and locally produced location estimates ($N = 100$)

estimate	error			
	mean	percentile		
		25	50	75
$\hat{\mathbb{L}}$	15.78 m	5.28 m	11.82 m	23.48 m
$\hat{\mathbb{L}}_{sh}$	16.23 m	6.77 m	13.05 m	23.35 m
$\hat{\mathbb{L}}_{oic}$	12.93 m	4.80 m	9.34 m	18.01 m
$\hat{\mathbb{L}}_{nic}$	17.92 m	8.25 m	15.19 m	25.61 m

again more notable with the increase in the number of nodes in the network, but the location accuracy improvements can be seen even in the smallest network simulation ($N = 20$).

The median error of the locally produced location estimate reduces slightly as the size of the network increases, again, due to the error converging towards a more stable value with the increase of the overall number of performed estimates. The simulation results show that using only locally produced location angle and distance estimates, a median error of under 12 m is achievable. Using estimate sharing, and making sure a locally produced estimate is part of the selected cluster of shared estimates, the median error of two-dimensional location estimate can be under 10 m, with a high node density, and under 13 m with a lower number of neighbours available for estimate sharing. The relative improvement of median errors for different number of nodes is shown in Table 4.19.

Table 4.19 Median location estimate error reduction: own to shared

estimate	median error		
	$N = 20$	$N = 50$	$N = 100$
$\hat{\mathbb{L}}$	13.58 m	12.76 m	11.82 m
$\hat{\mathbb{L}}_{oic}$	12.82 m	11.72 m	9.34 m
rel. imp.	0.94	0.92	0.79

With $N = 100$ nodes in the network, an improvement of over 20% of the location estimate error is achieved by estimate sharing.

Taking all of the results presented above into consideration, it is clear that the estimate sharing procedure is beneficial for the accuracy of the RNL algorithm. It is obvious that shared estimates should not be just taken by default, but that their efficiency is tightly related to the presence of a locally produced estimate in the cluster of chosen shared estimates. Bearing in mind the effects estimate sharing has on the estimated distance, a simple change in the estimate sharing procedure is proposed. This change assumes that once the location sharing process is complete, and the final shared estimate computed, the distance component of this estimate is substituted by the local distance estimate $\hat{d}$ (if available), and the two-dimensional location estimate is recalculated. This way, the overall accuracy of the final location estimate could increase.

In order to investigate this alternative approach, simulations including this small adjustment in the algorithm code were performed. The relative improvement of median location errors using the updated estimate sharing procedure is shown in Table 4.20. Figure 4.13 shows the step-by-step improvement of the location estimates, from own estimates to updated shared estimates.

The results of the repeated simulation test with the change described above show that the location estimate accuracy can be improved by further several percents compared to the original estimate sharing procedure, for the scenarios with low and medium node densities, while in the high node density scenario, there is no further increase. This result is in line with

Table 4.20 Median location estimate error reduction: own to shared; updated estimate sharing

estimate	median error		
	$N = 20$	$N = 50$	$N = 100$
$\hat{\mathbb{L}}$	15.12 m	10.57 m	12.28 m
$\hat{\mathbb{L}}_{oic}$	13.97 m	9.24 m	9.77 m
rel. imp.	0.92	0.87	0.79

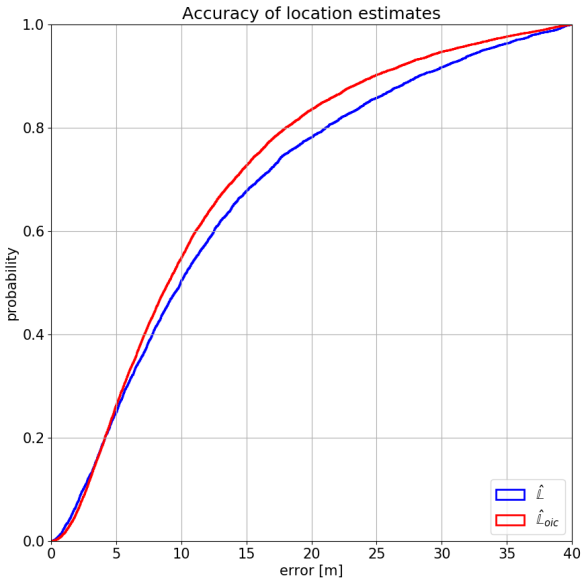


Fig. 4.13 Improvement of location estimates

the fact that the shared distance estimates in the high node density scenario are already very close to the ones locally produced by the RFR distance estimator.

4.5 Localisation Application

This section presents a potential application of the location estimates produced by the RNL algorithm in a *Wireless Sensor Network* (WSN) scenario. A distributed approach to sensing coverage control is proposed and the effects of the inaccuracy of the location estimates produced by the RNL algorithm is such a scenario are investigated.

Sensor networks represent one of the most important applications of wireless networks, especially infrastructureless ones. The principle they operate on is based on the assumption

of a group of nodes (sensors), connected through a wireless network, joining resources in order to work together on a common task. This task can, in general, be described as gathering relevant information about an area of interest and delivering this information to a dedicated sink. Each node gathers location-related data which can be in the form of measurements of certain physical phenomena, or images or video files. In the scenario considered throughout this thesis, involving *independently moving nodes* with *no access to any external localisation system*, efficient group location-based sensing is a challenge, as node distribution is both unknown and highly dynamic.

The network design parameters are usually dictated by the nature of the particular task the sensor network is dealing with. As a result, numerous different protocols and algorithms for group sensing have been developed over the years. One common goal of most of the existing solutions is the maintenance of a certain level of the geographical *coverage of the area of interest*, and their performance is measured by how efficiently it is achieved. Given such goal, it can be concluded that information about node locations is crucial in coverage control, which is the reason why it plays a big part in almost all of the existing solutions for efficient sensing coverage of the area of interest, regardless of the differences in scenarios and applications these solutions have been proposed for.

4.5.1 WSN coverage control

In static sensor networks solutions have been proposed along the lines of controlling the distribution of sensing nodes. Zhang et al. [108] show the relationship between coverage and connectivity in static sensor networks and propose a distributed density control algorithm. In [109] a non-uniform deterministic node distribution is proposed. The method of different subsets of sensing nodes being activated in different rounds to provide k-coverage of the field of interest is proposed in [110]. Multiple other solutions like the one proposed in [111],

require the nodes to be aware of their locations, either explicitly, or implicitly as is the case with deterministic node distribution.

Node location information is critical in mobile and hybrid sensing networks, allowing the node mobility to be utilised for network topology adjustments, thus maintaining coverage and connectivity, as proposed in [112, 113]. Paper [114] explores the virtual force approach, assuming that the movement of the sensing nodes is governed by the imaginary force exerted on them by other nodes or obstacles in the environment. In [115] mobile sensors are moved based on bids from static sensors, with coverage efficiency as the metric used for making movement decisions, while in [116] an algorithm for efficient path planning for a mobile sink node visiting designated static nodes is proposed. Mobile sensing algorithms in general work with the assumption that there is a terminal, stable, state towards which the nodes move. No nodes are allowed to move unless the change of location would improve sensing coverage. This results in node mobility only existing as a consequence of some changes in the network (after the initial network setup), without the sensing nodes being actively mobile.

While there are some solutions in the field of localisation-free systems, where nodes are unaware of their locations, the existing algorithms rely either on RSSI (Received Signal Strength Indicator) and hop-count based proximity estimates as is the case with [117]. In a network with dynamic topology, these metrics are unreliable, and solutions based on them cannot be expected to provide accurate results. Other localisation-free solutions propose random node sleeping patterns, as in [118, 119].

Without any location references, the only way to ensure a certain level of coverage of the area of interest would be to make all nodes send their data – resulting in a flooding-based mechanism – which is far from an efficient strategy. Furthermore, the design of WSNs assumes, in general, predetermined or controllable sensor distribution, often such that it is always possible to achieve complete coverage. In such scenarios, the coverage control problem becomes one of *finding the most efficient way to achieve complete coverage*.

In contrast, in a scenario characterised by independent node mobility and a lack of centralised control or communication, the coverage control problem becomes one of *finding the highest achievable (not necessarily complete) coverage, while doing it in an efficient way*.

4.5.2 Distributed coverage control

The sensing application used to evaluate the effects of using the location estimates produced by the RNL algorithm considers the same scenario of independently moving nodes communicating wirelessly through an ad hoc network. Each node is aware of its immediate neighbourhood, i.e. one-hop neighbours. It is worth pointing out that the *area of interest* is specific to each individual node, in line with the assumed distributed nature of the network. The area of interest of each node coincides with the node's communication range, as this is the area its neighbours can be located in.

As previously stated, the high dynamics and independent mobility of all nodes result in a system in which there can be no guarantee that complete coverage can be achieved, so the goal of each node is to learn the most about its area of interest, while still not resorting to a flooding approach when the number of available neighbours is high, so a certain level of efficiency is maintained. This efficiency is ensured by the process of selection of the neighbouring nodes whose sensing data is requested in one period, which is described in the following analysis.

In the interest of simplicity, the node trying to get coverage information from its neighbours will be referred to as the *root node*, and the neighbours selected to share their sensing data with the root node will be referred to as the *active neighbours*. In an ideal case, from the sensing coverage perspective, all nodes, including the root node and all its neighbours, would be distributed along the vertices of a triangular grid of dimension $a = r_s\sqrt{3}$, where r_s is the sensing range of each node, as shown in Figure 4.14. Such spacing of the nodes would

guarantee the optimal coverage, as there would be minimal overlap between individual node coverage areas, and there would be no gaps in coverage.

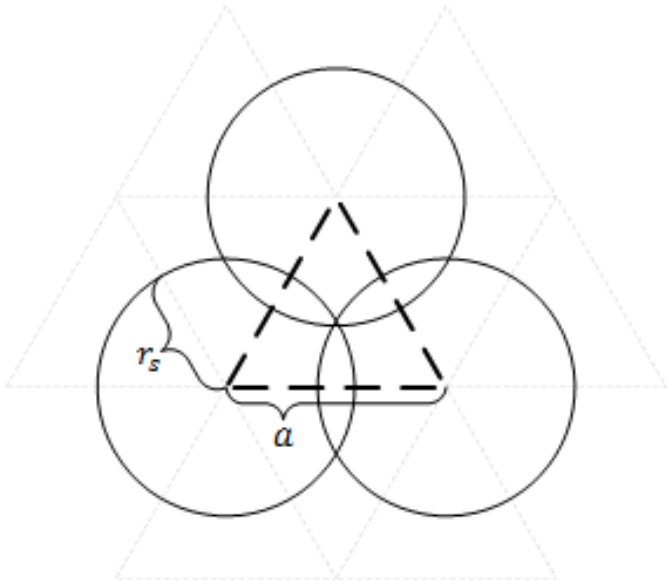


Fig. 4.14 Ideal triangular grid

In a realistic case, with non-deterministic neighbour distribution around the area of interest, there may be a large number of nodes whose sensing ranges are overlapping. In order to efficiently gather sensing data from the area of interest, the active neighbours are chosen using an empirical approach, which ensures that they are closest to the ideal neighbour locations. A set $\mathbb{G}$ of triangular grids across the area of interest is generated. Each grid has the same dimension $a = r_s\sqrt{3}$ with the root node as one of the vertices. The grids differ by the angle at which they are rotated with respect to the root node location, ranging from 0° to 60° (due to the symmetry of the grid), in $\frac{60°}{g}$ increments, where g is the total number of grids. The number of grids is a design parameter and represents the trade-off between the precision of the active neighbour selection procedure, and the length it may take to complete this procedure.

For each grid in $\mathbb{G}$, the following metric is calculated:

$$\sum_{j} \min_{n_i \in \mathbb{S}} d\left(v_j, n_i\right), \tag{4.15}$$

where v_j are vertices of the grid, and $d(v_j, n_i)$ represents the Euclidean distance between the position of the vertex j of the grid and the estimated location of the node n_i. The subset of nodes for which the lowest metric is obtained is the one which achieves the greatest coverage of the area of interest, and the nodes from this subset are selected active neighbours, i.e. ones that will share their sensing data with the root node in that period.

An example of the neighbour selection and resulting coverage is shown in Figure 4.15. Estimated relative locations of all seventeen neighbouring nodes are shown as crosses, next to the node numbers. The six nodes selected as active neighbours are shown in red circles, and the vertices of the grid for which the lowest metric from equation (4.15) was obtained are shown in blue circles. The communication range (area of interest radius) in this example is set to 30 m, while the sensing range is 15 m.

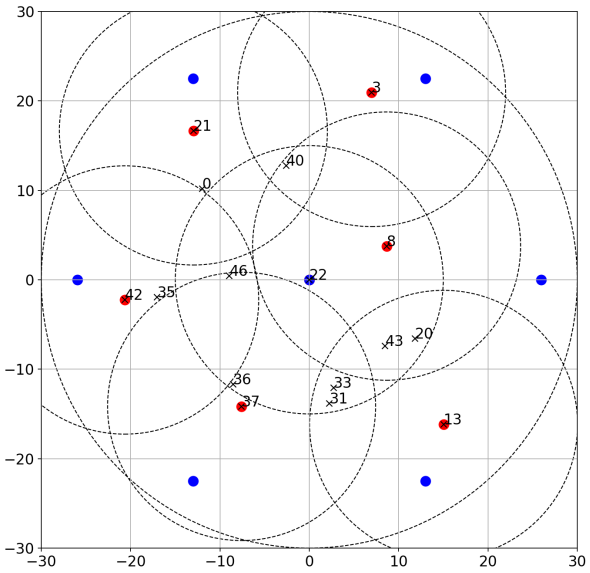


Fig. 4.15 Example of *active neighbour* selection

4.5.3 Results

The simulation setup used for evaluating coverage control based on RNL node location estimates is the same as the one used for the RNL performance evaluation, and is shown in Table 4.21. In addition to running the localisation algorithm at each step, each node now also applies the coverage control procedure described above on the location estimates. The following types of coverage are used in the coverage control performance analysis:

estimated coverage of the selected active neighbours based on their estimated locations

real coverage of the selected active neighbours based on their true locations

best optimal coverage achievable given the knowledge of the true locations of all neighbours

normalised ratio of real and best coverage

Table 4.21 Simulation parameters – coverage control

Parameter	Value	Unit
R	100	m
r_c	30	m
r_s	$\{10, 15\}$	m
(m_{min}, m_{max})	$(15, 20)$	s
N	$\{20, 50, 100\}$	-
dur	50	s

As in the previous simulation, the communication range is set to 30 m, which defines the area of interest of a particular node as a circle of a 30 m radius with the centre in the node itself. Three different node densities ($N = 20, N = 50$ and $N = 100$) are considered, and two different scenarios regarding the sensing range are examined, one smaller (10 m) and one greater (15 m). This means that a total of six distinct simulation tests have been performed.

The coverage results, using the values described above, are shown grouped by node density, in Tables 4.22, 4.23 and 4.24, for both the small and large sensing areas.

Table 4.22 Sensing coverage results ($N = 20$)

coverage	$r_s = 10\,\mathrm{m}$				$r_s = 15\,\mathrm{m}$			
	mean	percentile			mean	percentile		
		25	50	75		25	50	75
estimated	0.341	0.240	0.367	0.450	0.556	0.430	0.591	0.694
real	0.351	0.252	0.367	0.467	0.558	0.440	0.588	0.702
best	0.390	0.295	0.418	0.500	0.613	0.498	0.649	0.747
normalised	0.906	0.863	1.000	1.000	0.919	0.880	1.000	1.000

Table 4.23 Sensing coverage results ($N = 50$)

coverage	$r_s = 10\,\mathrm{m}$				$r_s = 15\,\mathrm{m}$			
	mean	percentile			mean	percentile		
		25	50	75		25	50	75
estimated	0.527	0.465	0.569	0.658	0.725	0.676	0.790	0.862
real	0.497	0.419	0.540	0.629	0.636	0.585	0.677	0.751
best	0.518	0.444	0.552	0.649	0.709	0.648	0.750	0.855
normalised	0.964	0.954	1.000	1.000	0.911	0.859	0.933	0.990

Table 4.24 Sensing coverage results ($N = 100$)

coverage	$r_s = 10\,\mathrm{m}$				$r_s = 15\,\mathrm{m}$			
	mean	percentile			mean	percentile		
		25	50	75		25	50	75
estimated	0.707	0.675	0.798	0.861	0.842	0.855	0.938	0.970
real	0.597	0.561	0.654	0.726	0.679	0.637	0.725	0.797
best	0.678	0.607	0.751	0.861	0.805	0.755	0.892	0.967
normalised	0.900	0.841	0.903	0.985	0.861	0.798	0.860	0.927

The values in the tables show the portion of the area of interest that is covered, i.e. is within a sensing area of one of the selected active neighbours in the given period.

The differences between the estimated and the real coverage are caused by the errors in the location estimates, seeing as the choice of active nodes is made based on the estimated positions. The real coverage achieved is almost always smaller than the estimated one, as the

true locations of the chosen nodes result in a greater overlap in sensing areas than the ones produced by the localisation algorithm.

The best coverage possible is, as expected, always greater than the real one, but the interesting results are the normalised ones, which show that, even with the inaccuracies in the location estimation procedure, these errors do not reduce the efficiency of the coverage control process critically, so the resulting coverage can be close to the best possible one for the given scenario. The effects of this are best seen when a higher number of nodes is present in the network, as there are more possibilities of choosing a wrong node. On the other hand, the best coverage achievable depends highly on the node density and the sensing range, which is expected.

4.6 Location Tracking

As described in section 3.7, the accuracy of location estimates can be improved by applying location tracking over time, during periods, or sequences, of constant relative mobility parameter values. This improvement has been applied on a single simulation run with the same parameters as shown in Table 4.7. The procedure of location tracking has been applied to raw location estimates produced by the RNL, for sequences which contained more than two estimates, and for which the ambiguity has been resolved. Figure 4.16 shows the overall reduction of location estimate error achieved by applying location tracking, as the distribution of errors is considerably shifted toward the smaller values.

Location tracking is best described using a step-by-step view of the procedure. Figure 4.17 shows the location tracking procedure at different steps. In each of the figures representing a single step in a sequence of constant relative mobility of the target node, the blue dots describe the raw RNL estimates and the true relative location of the target node is shown in

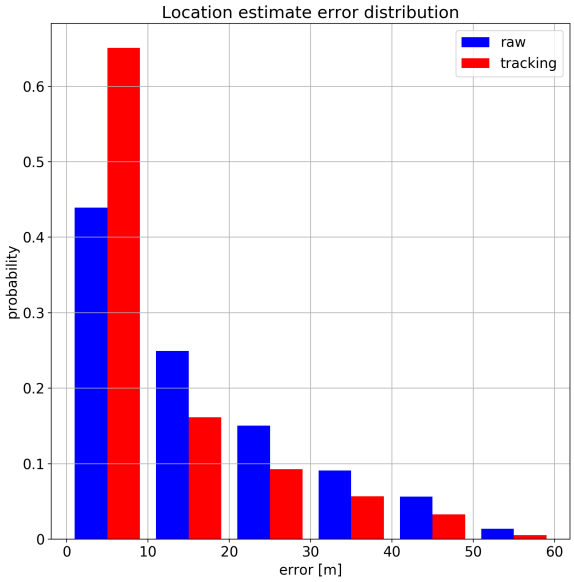
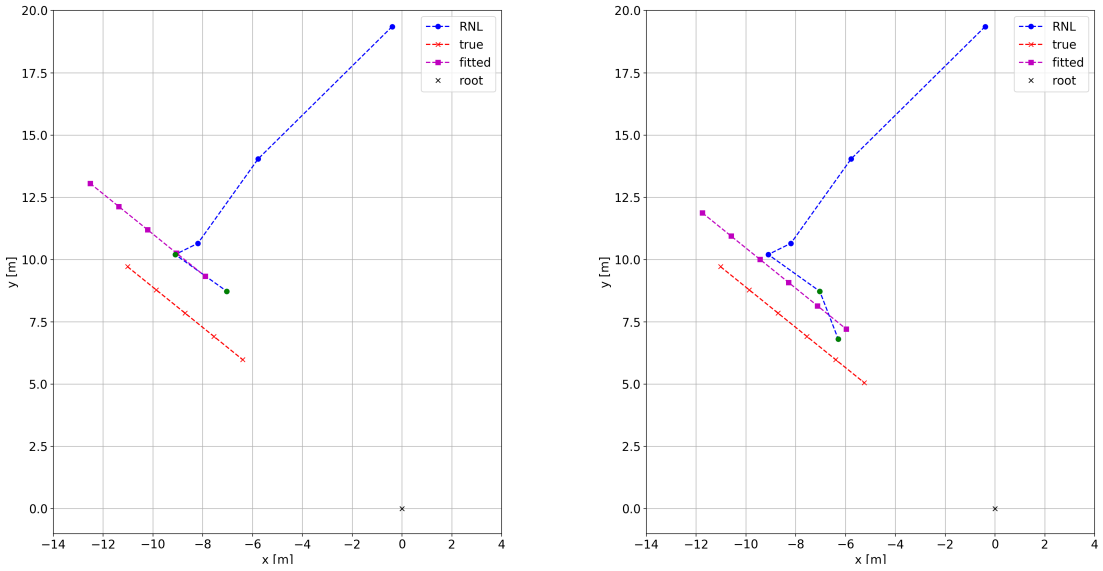


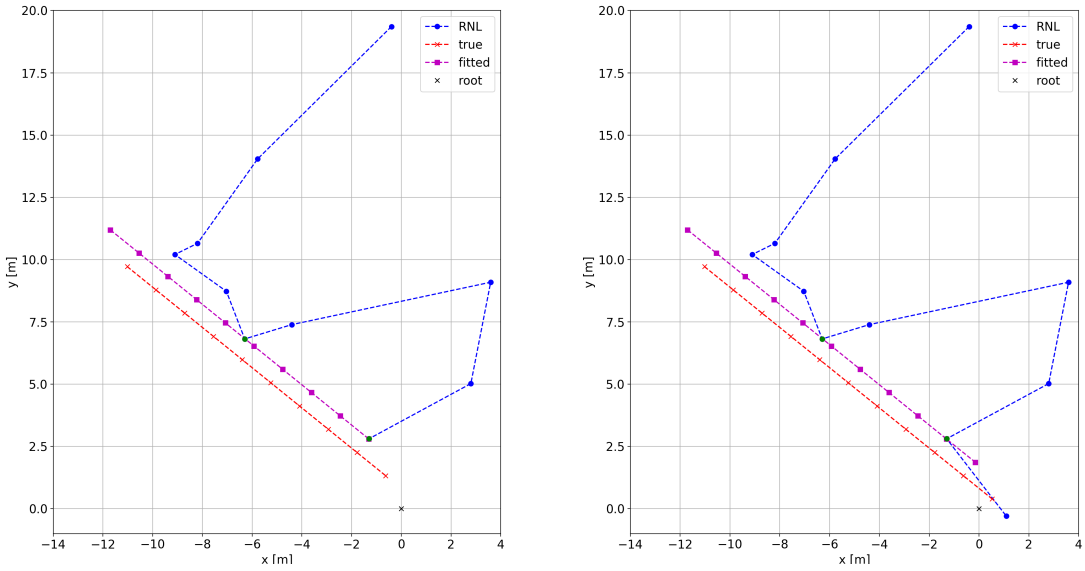
Fig. 4.16 Location estimate error distribution

red crosses. Following the procedure described in section 3.7 at each step, a pair of anchor points are chosen, shown in green, based on which the fitted line of corrected estimates is drawn, shown in purple.



(a) Step 4 (b) Step 5

Fig. 4.17 Location tracking procedure



(c) Step 9 (d) Step 10

Fig. 4.17 Location tracking procedure (cont.)

These corrections are dynamic, and the improvement in location estimate accuracy they enable can increase over time. This can be noted comparing the raw and the fitted location estimates to the true location over time, as the errors with great magnitude have been significantly corrected. Such results are in line with the ones shown in Figure 4.16, where the biggest difference in raw and corrected estimates is the considerable increase of the proportion of errors under 10 m.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis presented a distributed localisation solutions for ad hoc networks of moving nodes without access to external localisation signals or systems. The proposed *Relative Neighbour Localisation* (RNL) algorithm is based on the relationship between the trends in signal strength measurements and the mobility parameters on one side, and the relative neighbour location on the other. The algorithm enables mobile nodes in ad hoc networks to produce location estimates of their one-hop neighbours, without relying on any additional localisation infrastructure.

Ranging methods for localisation are used in networks where the capability of performing precise measurements exists either by the network infrastructure or specific ranging devices. The use of reference points is possible when infrastructure is in place, or some nodes are assumed to somehow be aware of their location. Network-wide information exchange can be used if the network topology is static enough that it does not change in time information gathering and processing, as well as the dissemination of the results, is performed. None of these approaches is applicable to an ad hoc network of moving nodes without access to external localisation systems.

The basis of the proposed solution is the intuitive relationship between the mobility of the neighbouring node, the strength of the signal coming from it, and the relative location of that node. An ideal case localisation principle has been analytically derived, and the empirical application of this principle to a realistic scenario was described. The localisation algorithm is formed of blocks for angle and distance estimation, as well as an additional estimate sharing block. The operation of each of the blocks and the computation of the final location estimates were presented in detail.

In order to evaluate the performance of the proposed solution, a simulation environment has been developed. The simulation models are calibrated based on real-life experimental results gathered from an experimental testbed. The models were shown to be consistent with the experimental measurements, thus allowing for realistic large-scale tests to be conducted using the simulation. It was shown how different steps in the process of angle estimation improve the angle estimate accuracy, as well as how estimate sharing is beneficial to the accuracy of the final two-dimensional location estimate. A median location estimate error of under 10 m was achieved, while the median location angel estimate error was shown to be around 35°. The location estimates produced by the RNL were applied to a distributed sensing scenario which was designed. The coverage of the area of interest obtained based on the RNL estimates was shown to be over 90% of the one achievable in the case true locations of all of the neighbouring nodes were known.

5.2 Future Work

Several paths for further improvements or application of the RNL algorithm exist:

- ~~Location estimates are produced by the RNL algorithm in a step-by-step manner, where each estimate is completely independent of any previous ones. The nodes, on the other hand, do not move in that way, their locations over time are correlated.~~

- Processing of the location estimates, either by angle and distance components or as two-dimensional points, could help improve the accuracy of these estimates. The time series of the location estimates could be fitted into the analytical form of the expected values of these estimates based on the laws of physics.

- Based on the processing mentioned above, individual estimates can be assigned a certainty value. Detection of clear peaks in signal strength measurements can be used as an indicator to the relative location of the neighbour, which can be used to reinforce some estimates.

- All estimates received through the process of location sharing are treated equally. Should a certainty-based weighing approach be implemented, different estimates could participate in the shared location procedure proportionally to their certainty.

- Different application may require less precise measurements, but with higher certainty. The process of estimate sharing could be expended in such way that it introduces additional constraints for discarding some estimates as impossible, leaving the final estimate as a general area or an angle sector in which the node is located with high certainty.

References

[1] Aboelmagd Noureldin. *Global Positioning System*, volume 7 of *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*, pages 65–123. Springer-Verlag, US, 2006.

[2] Bernhard Hofmann-Wellenhof, Herbert Lichtenegger, and Elmar Wasle. *GLONASS*, chapter 10, pages 341–364. GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more. Springer Science & Business Media, 2007.

[3] John M. Dow, Ruth E. Neilan, and Chris Rizos. The international GNSS service in a changing landscape of global navigation satellite systems. *Journal of geodesy*, 83(3-4):191–198, 2009.

[4] Geoffrey Blewitt. *Basics of the GPS technique: Observation Equations*, pages 10–54. Geodetic applications of GPS. Nordic Geodetic Commission Sweden, 1997.

[5] Ka W. Cheung, Hing-Cheung So, W-K Ma, and Yiu-Tong Chan. Least squares algorithms for time-of-arrival-based mobile location. *IEEE Transactions on Signal Processing*, 52(4):1121–1130, 2004.

[6] Sreeram Potluri. Hyperbolic position location estimator with TDOAS from four stations. Master's thesis, NJIT, 2002.

[7] Peng Rong and Mihail L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *2006 3rd annual IEEE communications society on sensor and ad hoc communications and networks*, volume 1, pages 374–382. Ieee, 2006.

[8] Federal Communications Commission. Wireless E911 Location Accuracy Requirements. https://apps.fcc.gov/edocs_public/attachmatch/FCC-15-9A1_Rcd.pdf, Feb. 2015. Accessed: 2019-02-12.

[9] Gökhan Kul, Tansel Özyer, and Bülent Tavli. Ieee 802.11 wlan based real time indoor positioning: Literature survey and experimental investigations. *Procedia Computer Science*, 34:157–164, 2014.

[10] 3GPP. Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 LCS specification Location Services (LCS); (GSM 04.71 version 7.0.0 Release 1998). Technical report, 3GPP, Aug 1999.

[11] 3GPP. Universal Mobile Telecommunications System (UMTS); Stage 2 functional specification of UE positioning in UTRAN (3GPP TS 25.305 version 3.7.0 Release 1999). Technical report, 3GPP, Dec 2001.

[12] 3GPP. LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Positioning Protocol (LPP) (3GPP TS 36.355 version 9.4.0 Release 9). Technical report, 3GPP, Jan 2011.

[13] Emiliano Trevisani and Andrea Vitaletti. Cell-ID location technique, limits and benefits: an experimental study. In *Mobile computing systems and applications, 2004. WMCSA 2004. Sixth IEEE workshop on*, pages 51–60. IEEE, 2004.

[14] Maurizio A. Spirito, Sami Poykko, and Olli Knuuttila. Experimental performance of methods to estimate the location of legacy handsets in GSM. In *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, volume 4, pages 2716–2720. IEEE, 2001.

[15] Ding-Bing Lin and Rong-Terng Juang. Mobile location estimation based on differences of signal attenuations for GSM systems. *IEEE transactions on vehicular technology*, 54(4):1447–1454, 2005.

[16] Yilin Zhao. Mobile phone location determination and its impact on intelligent transportation systems. *IEEE Transactions on intelligent transportation systems*, 1(1):55–64, 2000.

[17] Sven Fischer, Havish Koorapaty, Erik Larsson, and Ari Kangas. System performance evaluation of mobile positioning methods. In *Vehicular Technology Conference, 1999 IEEE 49th*, volume 3, pages 1962–1966. IEEE, 1999.

[18] Motorola. Time Aligned IP-DL positioning technique. Technical report, TSG-RAN Working Group 1, Ad Hoc 17, Meeting 7, September 1999.

[19] Jakub Borkowski, Jarno Niemelä, and Jukka Lempiäinen. Performance of cell ID RTT hybrid positioning method for UMTS radio networks. In *Proceedings of the 5th European Wireless Conference*, pages 487–492, 2004.

[20] Jakub Borkowski, Jamo Niemela, and Jukka Lempiainen. Enhanced performance of Cell ID RTT by implementing forced soft handover algorithm. In *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*, volume 5, pages 3545–3549. IEEE, 2004.

[21] Jakub Borkowski and Jukka Lempiäinen. Practical network-based techniques for mobile positioning in UMTS. *EURASIP Journal on Applied Signal Processing*, 2006:149, 2006.

[22] José A. del Peral-Rosado, José A. López-Salcedo, Gonzalo Seco-Granados, Francesca Zanier, and Massimo Crisci. Achievable localization accuracy of the positioning reference signal of 3gpp lte. In *2012 International Conference on Localization and GNSS*, pages 1–6. IEEE, 2012.

[23] Jonas Medbo, Iana Siomina, Ari Kangas, and Johan Furuskog. Propagation channel impact on lte positioning accuracy: A study based on real measurements of observed time difference of arrival. In *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pages 2213–2217. IEEE, 2009.

[24] Wen Xu, Ming Huang, Chen Zhu, and Armin Dammann. Maximum likelihood TOA and OTDOA estimation with first arriving path detection for 3GPP LTE system. *Transactions on Emerging Telecommunications Technologies*, 27(3):339–356, 2016.

[25] Reza M. Vaghefi and R. M. Buehrer. Improving positioning in LTE through collaboration. In *Positioning, Navigation and Communication (WPNC), 2014 11th Workshop on*, pages 1–6. IEEE, 2014.

[26] TehnoCom. TruePosition Indoor Test Report Wilmington, DE. https://ecfsapi.fcc.gov/file/7521337391.pdf, 2014. Accessed: 2019-01-15.

[27] Marco Driusso, Chris Marshall, Mischa Sabathy, Fabian Knutti, Heinz Mathis, and Fulvio Babich. Vehicular Position Tracking Using LTE Signals. *IEEE Trans.Vehicular Technology*, 66(4):3376–3391, 2017.

[28] Christian Gentner, Estefanía Muñoz, Mohammed Khider, Emanuel Staudinger, Stephan Sand, and Armin Dammann. Particle filter based positioning with 3GPP-LTE in indoor environments. In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pages 301–308. IEEE, 2012.

[29] José A. del Peral-Rosado, Michele Bavaro, José A. López-Salcedo, Gonzalo Seco-Granados, Pravir Chawdhry, Joaquim Fortuny-Guasch, Paolo Crosta, Francesca Zanier, and Massimo Crisci. Floor detection with indoor vertical positioning in LTE femtocell networks. In *Globecom Workshops (GC Wkshps), 2015 IEEE*, pages 1–6. IEEE, 2015.

[30] Po-Hsuan Tseng and Ke-Ting Lee. A femto-aided location tracking algorithm in LTE-A heterogeneous networks. *IEEE Transactions on Vehicular Technology*, 66(1):748–762, 2017.

[31] Frank Stephen Tromp Van Diggelen. *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House, 2009.

[32] Jeongyeup Paek, Kyu-Han Kim, Jatinder P. Singh, and Ramesh Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 293–306. ACM, 2011.

[33] Anthony J. Weiss. On the accuracy of a cellular location system based on rss measurements. *IEEE transactions on vehicular technology*, 52(6):1508–1518, 2003.

[34] Hui Zang, Francois Baccelli, and Jean Bolot. Bayesian inference for localization in cellular networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[35] Heikki Laitinen, Jaakko Lahteenmaki, and Tero Nordstrom. Database correlation method for gsm location. In *Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd*, volume 4, pages 2504–2508. IEEE, 2001.

[36] Veljo Otsason, Alex Varshavsky, Anthony LaMarca, and Eyal De Lara. Accurate GSM indoor localization. In *International conference on ubiquitous computing*, pages 141–158. Springer, 2005.

[37] Mohamed Ibrahim and Moustafa Youssef. CellSense: An accurate energy-efficient GSM positioning system. *IEEE Transactions on Vehicular Technology*, 61(1):286–296, 2012.

[38] Yao-hui Wang, Hu Li, Xin-long Luo, Qi-ming Sun, and Jin-nan Liu. A 3D fingerprinting positioning method based on cellular networks. *International Journal of Distributed Sensor Networks*, 10(7):248981, 2014.

[39] Ayon Chakraborty, Luis E. Ortiz, and Samir R. Das. Network-side positioning of cellular-band devices with minimal effort. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 2767–2775. IEEE, 2015.

[40] Torbjörn Wigren. Adaptive enhanced cell-id fingerprinting localization by clustering of precise position measurements. *IEEE Transactions on Vehicular Technology*, 56(5):3199–3209, 2007.

[41] Jenni Wennervirta and Torbjörn Wigren. RTT positioning field performance. *IEEE Transactions on Vehicular Technology*, 59(7):3656–3661, 2010.

[42] Torbjörn Wigren. Fingerprinting localisation using round trip time and timing advance. *IET communications*, 6(4):419–427, 2012.

[43] Torbjörn Wigren. LTE Fingerprinting Localization with Altitude. In *Vehicular Technology Conference, IEEE 38th*, pages 1–5, 2012.

[44] Arvind Thiagarajan, Lenin Ravindranath, Hari Balakrishnan, Samuel Madden, and Lewis Girod. Accurate, Low-energy Trajectory Mapping for Mobile Devices. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, pages 267–280. USENIX Association, 2011.

[45] Jussi Turkka and Tapani Ristaniemi. Deliverable D6.3 Localization architecture for multi-layer, multi-RAT heterogeneous network. Technical report, 2015.

[46] Reza M. Vaghefi and R. M. Buehrer. Cooperative RF pattern matching positioning for LTE cellular systems. In *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*, pages 264–269. IEEE, 2014.

[47] Xinrong Li, Kaveh Pahlavan, Matti Latva-aho, and Mika Ylianttila. Comparison of indoor geolocation methods in DSSS and OFDM wireless LAN systems. In *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No. 00CH37152)*, volume 6, pages 3015–3020. IEEE, 2000.

[48] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, pages 173–184. ACM, 2010.

[49] Henri Nurminen, Jukka Talvitie, Simo Ali-Löytty, Philipp Müller, Elena-Simona Lohan, Robert Piché, and Markku Renfors. Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks. In *2012*

International Conference on Indoor Positioning and Indoor Navigation (IPIN), pages 1–9. IEEE, 2012.

[50] Luigi Bruno, Mohammed Khider, and Patrick Robertson. On-line training of the path-loss model in Bayesian WLAN indoor positioning. In *International Conference on Indoor Positioning and Indoor Navigation*, pages 1–9. IEEE, 2013.

[51] Suining He, Tianyang Hu, and S-H G. Chan. Contour-based trilateration for indoor fingerprinting localization. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 225–238. ACM, 2015.

[52] Paramvir Bahl and Venkata N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *IEEE infocom*, volume 2, pages 775–784. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2000.

[53] Paramvir Bahl, Venkata N. Padmanabhan, and Anand Balachandran. Enhancements to the RADAR user location and tracking system. *Microsoft Research*, 2(MSR-TR-2000-12):775–784, 2000.

[54] Moustafa A. Youssef, Ashok Agrawala, and A. U. Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.(PerCom 2003).*, pages 143–150. IEEE, 2003.

[55] Moustafa Youssef and Ashok Agrawala. The Horus WLAN location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 205–218. ACM, 2005.

[56] Simon Yiu, Marzieh Dashti, Holger Claussen, and Fernando Perez-Cruz. Wireless rssi fingerprinting localization. *Signal Processing*, 131:235–244, 2017.

[57] Genming Ding, Zhenhui Tan, Jinsong Wu, and Jinbao Zhang. Efficient indoor fingerprinting localization technique using regional propagation model. *IEICE Transactions on Communications*, 97(8):1728–1741, 2014.

[58] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Intelligent dynamic radio tracking in indoor wireless local area networks. *IEEE Transactions on Mobile Computing*, 9(3):405–419, 2010.

[59] Anthea W. S. Au, Chen Feng, Shahrokh Valaee, Sophia Reyes, Sameh Sorour, Samuel N. Markowitz, Deborah Gold, Keith Gordon, and Moshe Eizenman. Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device. *IEEE Transactions on Mobile Computing*, 12(10):2050–2062, 2013.

[60] Genming Ding, Zhenhui Tan, Jinsong Wu, Jinshan Zeng, and Lingwen Zhang. Indoor fingerprinting localization and tracking system using particle swarm optimization and Kalman filter. *IEICE Transactions on Communications*, 98(3):502–514, 2015.

[61] Suining He, S-H G. Chan, Lei Yu, and Ning Liu. Fusing noisy fingerprints with distance bounds for indoor localization. In *2015 IEEE conference on computer communications (INFOCOM)*, pages 2506–2514. IEEE, 2015.

[62] Adriano Moreira, Ivo Silva, Filipe Meneses, Maria J. Nicolau, Cristiano Pendao, and Joaquín Torres-Sospedra. Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2017.

[63] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2004.

[64] Christos Laoudias, Robert Piché, and Christos G. Panayiotou. Device self-calibration in location systems using signal strength histograms. *Journal of Location Based Services*, 7(3):165–181, 2013.

[65] Yungeun Kim, Hyojeong Shin, Yohan Chon, and Hojung Cha. Smartphone-based Wi-Fi tracking system exploiting the RSS peak to overcome the RSS variance problem. *Pervasive and Mobile Computing*, 9(3):406–420, 2013.

[66] Gough Lui, Thomas Gallagher, Binghao Li, Andrew G. Dempster, and Chris Rizos. Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization. In *2011 International Conference on Localization and GNSS (ICL-GNSS)*, pages 53–57. IEEE, 2011.

[67] Joaquín Torres-Sospedra, Raúl Montoliu, Sergio Trilles, Oscar Belmonte, and Joaquín Huerta. Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems. *Expert Systems with Applications*, 42(23):9263–9278, 2015.

[68] Joaquín Torres-Sospedra and Adriano Moreira. Analysis of sources of large positioning errors in deterministic fingerprinting. *Sensors*, 17(12):2736, 2017.

[69] Nirupama Bulusu, John Heidemann, and Deborah Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE personal communications*, 7(5):28–34, 2000.

[70] Lance Doherty and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, volume 3, pages 1655–1663. IEEE, 2001.

[71] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek F. Abdelzaher. Range-free localization and its impact on large scale sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 4(4):877–906, 2005.

[72] Jang-Ping Sheu, Pei-Chun Chen, and Chih-Shun Hsu. A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement. *IEEE transactions on mobile computing*, 7(9):1110–1123, 2008.

[73] Dragos Niculescu and Badri Nath. Ad hoc positioning system (APS). In *GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)*, volume 5, pages 2926–2931. IEEE, 2001.

[74] Yun Wang, Xiaodong Wang, Demin Wang, and Dharma P. Agrawal. Range-free localization using expected hop progress in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(10):1540–1552, 2009.

[75] Hyunjae Woo, Sangwoo Lee, and Chaewoo Lee. Range-free localization with isotropic distance scaling in wireless sensor networks. In *The International Conference on Information Networking 2013 (ICOIN)*, pages 632–636. IEEE, 2013.

[76] Ahmad El Assaf, Slim Zaidi, Sofiene Affes, and Nahi Kandil. Range-free localization algorithm for heterogeneous wireless sensor networks. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2805–2810. IEEE, 2014.

[77] C. S. J. Rabaey and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX technical annual conference*, pages 317–327, 2002.

[78] Radhika Nagpal, Howard Shrobe, and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *Information processing in sensor networks*, pages 333–348. Springer, 2003.

[79] Jizhong Zhao, Wei Xi, Yuan He, Yunhao Liu, Xiang-Yang Li, Lufeng Mo, and Zheng Yang. Localization of wireless sensor networks in the wild: Pursuit of ranging quality. *IEEE/ACM Transactions on Networking (ToN)*, 21(1):311–323, 2013.

[80] Srdjan Čapkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-free positioning in mobile ad hoc networks. *Cluster Computing*, 5(2):157–167, 2002.

[81] Andreas Savvides, Heemin Park, and Mani B. Srivastava. The bits and flops of the n-hop multilateration primitive for node localization problems. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 112–121. ACM, 2002.

[82] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 201–212. ACM, 2003.

[83] Xiang Ji and Hongyuan Zha. Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling. In *IEEE INFOCOM 2004*, volume 4, pages 2652–2661. IEEE, 2004.

[84] Pratik Biswas and Yinyu Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

[85] Tzu-Chen Liang, Ta-Chung Wang, and Yinyu Ye. A gradient search method to round the semidefinite programming relaxation solution for ad hoc wireless sensor network localization. *Sanford University, formal report*, 5, 2004.

[86] Anushiya A. Kannan, Guoqiang Mao, and Branka Vucetic. Simulated annealing based localization in wireless sensor network. In *The IEEE Conference on Local Computer Networks 30th Anniversary (LCN'05) l*, page 514. IEEE, 2005.

[87] Anushiya A. Kannan, Guoqiang Mao, and Branka Vucetic. Simulated annealing based wireless sensor network localization with flip ambiguity mitigation. In *2006 IEEE 63rd Vehicular Technology Conference*, volume 2, pages 1022–1026. IEEE, 2006.

[88] Sangwoo Lee, Hyunjae Woo, and Chaewoo Lee. Wireless sensor network localization with connectivity-based refinement using mass spring and kalman filtering. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):152, 2012.

[89] Yi Shang and Wheeler Ruml. Improved MDS-based localization. In *IEEE INFOCOM 2004*, volume 4, pages 2640–2651. IEEE, 2004.

[90] Mo Li and Yunhao Liu. Rendered path: range-free localization in anisotropic sensor networks with holes. *IEEE/ACM Transactions on Networking (ToN)*, 18(1):320–332, 2010.

[91] Bin Xiao, Lin Chen, Qingjun Xiao, and Minglu Li. Reliable anchor-based sensor localization in irregular areas. *IEEE Transactions on Mobile Computing*, 9(1):60–72, 2010.

[92] Qingjun Xiao, Bin Xiao, Jiannong Cao, and Jianping Wang. Multihop range-free localization in anisotropic wireless sensor networks: A pattern-driven scheme. *IEEE Transactions on Mobile Computing*, 9(11):1592–1607, 2010.

[93] Xuan Liu, Shigeng Zhang, Jianxin Wang, Jiannong Cao, and Bin Xiao. Anchor supervised distance estimation in anisotropic wireless sensor networks. In *2011 IEEE Wireless Communications and Networking Conference*, pages 938–943. IEEE, 2011.

[94] Sangwoo Lee, Bonhyun Koo, and Sunwoo Kim. Raps: reliable anchor pair selection for range-free localization in anisotropic networks. *IEEE Communications Letters*, 18(8):1403–1406, 2014.

[95] Sangwoo Lee, Myungjun Jin, Bonhyun Koo, Cheonsig Sin, and Sunwoo Kim. Pascal's triangle-based range-free localization for anisotropic wireless networks. *Wireless Networks*, 22(7):2221–2238, 2016.

[96] H.T. Friis. Simple Transmission Formula. *Proceedings of the IRE*, 34(5):254–256, 1946.

[97] Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang, and Chenglong Wang. Distance measurement model based on RSSI in WSN. *Wireless Sensor Network*, 2(08):606, 2010.

[98] Paolo Pivato, Luigi Palopoli, and Dario Petri. Accuracy of RSS-based centroid localization algorithms in an indoor environment. *IEEE Transactions on Instrumentation and Measurement*, 60(10):3451–3460, 2011.

[99] Omotayo G. Adewumi, Karim Djouani, and Anish M. Kurien. RSSI based indoor and outdoor distance estimation for localization in WSN. In *2013 IEEE international conference on Industrial technology (ICIT)*, pages 1534–1539. IEEE, 2013.

[100] Joe H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[101] Travis E. Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.

[102] Eben Upton and Gareth Halfacree. *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[103] Raspberry Pi 2 Model B. https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. Accessed: 2019-02-27.

[104] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *Tree-based methods*, pages 303–335. An Introduction to Statistical Learning. Springer, 2013.

[105] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[106] Naomi S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[107] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *Support vector machines*, pages 337–372. An Introduction to Statistical Learning. Springer, 2013.

[108] Honghai Zhang and Jennifer C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):89–124, 2005.

[109] Xiaobing Wu, Guihai Chen, and Sajal K. Das. Avoiding energy holes in wireless sensor networks with nonuniform node distribution. *IEEE Transactions on Parallel and Distributed Systems*, 19(5):710–720, 2008.

[110] Habib M. Ammari and Sajal K. Das. Centralized and clustered k-coverage protocols for wireless sensor networks. *IEEE Transactions on Computers*, 61(1):118–133, 2012.

[111] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39. ACM, 2003.

[112] Guiling Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.

[113] Saurabh Ganeriwal, Aman Kansal, and Mani B. Srivastava. Self aware actuation for fault repair in sensor networks. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 5, pages 5244–5249. IEEE, 2004.

[114] Nojeong Heo and Pramod K. Varshney. An intelligent deployment and clustering algorithm for a distributed mobile sensor network. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, volume 5, pages 4576–4581. IEEE, 2003.

[115] Guiling Wang, Guohong Cao, Piotr Berman, and Thomas F. La Porta. Bidding protocols for deploying mobile sensors. *IEEE Transactions on Mobile Computing*, 6(5):563–576, 2007.

[116] Hamidreza Salarian, Kwan-Wu Chin, and Fazel Naghdy. An energy-efficient mobile-sink path selection strategy for wireless sensor networks. *IEEE Transactions on vehicular technology*, 63(5):2407–2419, 2014.

[117] Bang Wang, Cheng Fu, and Hock B. Lim. Layered diffusion-based coverage control in wireless sensor networks. *Computer Networks*, 53(7):1114–1124, 2009.

[118] Santosh Kumar, Ten H. Lai, and József Balogh. On k-coverage in a mostly sleeping sensor network. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 144–158. ACM, 2004.

[119] Fan Ye, Gary Zhong, Jesse Cheng, Songwu Lu, and Lixia Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *23rd International Conference on Distributed Computing Systems, 2003. Proceedings.*, pages 28–37. IEEE, 2003.